

UNITED STATES PATENT APPLICATION

for

METHOD AND SYSTEM FOR PREVENTING UNAUTHORIZED RECORDING OF
MEDIA CONTENT ON A MACINTOSH OPERATING SYSTEM

Inventors:

HANK RISAN
EDWARD VINCENT FITZGERALD

Prepared by:

WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, CA 95113
(408) 938-9060

METHOD AND SYSTEM FOR PREVENTING UNAUTHORIZED RECORDING OF MEDIA CONTENT ON A MACINTOSH OPERATING SYSTEM

FIELD OF THE INVENTION

5 The present invention relates to the recording of media content. More particularly, the present invention relates to preventing unauthorized recording of media content on a Macintosh operating system.

BACKGROUND OF THE INVENTION

10 With advancements in hardware and software technology, computers are integral tools utilized in various applications, such as finance, CAD (computer aided design), manufacturing, health care, telecommunication, education, etc. Further, an enhancement in computer functionality can be realized by communicatively coupling computers together to form a network. Within a network environment, computer systems enable
15 users to exchange files, share information stored in common databases, combine or pool resources, communicate via electronic mail (e-mail), and access information on the Internet. Additionally, computers connected to a network environment, e.g., the Internet, provide their users access to data and information from all over the world.

20 Some of the various types of data that a user can access and share include, but are not limited to, text data such as that found in a word document, graphical data such as that found in pictures, e.g., JPEGs, GIFs, TIFFs, audio data such as that found in music files, e.g., MP3 files, and video data such as that found in moving pictures files, e.g., MPEG, MOV, and AVI files, to name a few. In fact, nearly any type of data can be

stored and shared with other computer systems. In many instances, the material contained within the various data types is copyrighted material.

There are many different types of network environments that can be implemented to facilitate sharing of data between computer systems. Some of the various network environment types include Ethernet, client-server, and wired and/or wireless network environments. A common utilization of a network environment type is for file sharing, such as in a P2P network or point-to-point network. Most P2P networks rely on business models based upon the transfer and redistribution of copyrighted material, e.g., audio files, between computers coupled to a network, e.g., the Internet. A P2P network allows a user to acquire the copyrighted material from a computer, a web site source, or a music broadcaster, and store and share the material with other users throughout the network, in some instances acting as a web site source or a music broadcaster.

15

It is also common for users sharing media files in an uncontrolled manner to use freely distributed or commercially available media player applications to experience, e.g., listen, view, and/or watch, the shared files. In many instances, these media player applications also provide for downloading the media file from a P2P network or from licensed web broadcasters, saving it locally, and then upload the media file onto an unlawful P2P or similar network and/or consumer recording devices. Unlawfully saving/recording a media file can be as simple as selecting the save or record function on a media player application.

20

Additionally, many of the computers, web sites, and web broadcasters that share copyrighted material commonly do not control or monitor the files being exchanged between computers. Additionally, when web sites attempt to control or restrict the distribution of copyrighted material, e.g., audio files, users seeking to circumvent controls or restrictions can, in many cases, simply utilize the recording functionality of a media player application and save the copyrighted material, rename the particular audio file, and upload the renamed file, rendering attempts to control or restrict its distribution moot.

Further, many of the media player/recorder applications are designed to capture and record incoming media files in a manner that circumvents controls implemented by a media player application inherent to an operating system, e.g., QuickTime for Apple, MediaPlayer for Windows™, etc., or one downloadable from the Internet, e.g., RealPlayer, LiquidAudio, or those provided by webcasters, e.g., PressPlay, for controlling unauthorized recording of media files. Additionally, many digital recording devices, e.g., mini-disc recorders, MP3 recorders, and the like, can be coupled to a digital output of a computer system to capture the media file.

It is desired to prevent persons from making unauthorized copies of copyrighted material through some available network, e.g., wireline, wireless, P2P, etc., or through a communicative coupling. It is further desirable to prevent persons from making unauthorized copies of media files from or to alternative sources, e.g., CD players, DVD players, removable hard drives, personal electronic and/or recording devices, e.g., MP3 recorders, and the like.

Current methods of sharing media files do not provide adequate protection against unauthorized recording of the media files.

SUMMARY OF THE INVENTION

Accordingly, a need exists for a method that prevents unauthorized recording of media files on a Macintosh operating system. Further, a need exists for a method that prevents unauthorized recording of media files on a Macintosh operating system that can be updated over a network. Yet another need exists for a method that prevents unauthorized recording of media files on a Macintosh operating system which is compatible with industry standards. Embodiments of the present invention satisfy the above-mentioned needs.

In one embodiment, the present invention provides a method for preventing unauthorized recording of media content on a Macintosh operating system. The present method registers a compliance mechanism on a client system having the Macintosh operating system operating thereon. The compliance mechanism comprises a framework for validating the compliance mechanism on the client system, and a multimedia component opened by the framework. The present method uses the multimedia component for decrypting the media content on the client system. The present method also prevents decryption of the media content on the client system having the Macintosh operating system operating thereon if a portion of the compliance mechanism is invalidated.

In another embodiment, the present invention provides computer implementable instructions stored on a computer readable medium, the instructions for causing a client system to perform a method for preventing unauthorized recording of media content on a Macintosh operating system. The present method registers a compliance mechanism on a client system having the Macintosh operating system operating

thereon. The compliance mechanism comprises a framework for validating the compliance mechanism on the client system, a multimedia component opened by the framework, and a kernel level extension communicatively coupled with the framework. If the framework detects an invalid environment, the multimedia component will

5 prevent decryption of the media content on the client system. The kernel level extension provides kernel level driver information to the framework such that the output of the media content on the client system having the Macintosh operating system operating thereon is disabled if a portion of the compliance mechanism is invalidated.

10

In another embodiment, the present invention provides a system for preventing unauthorized recording of media content on a Macintosh operating system. The present method provides a means for registering a compliance mechanism on a client system having the Macintosh operating system operating thereon. The compliance

15 mechanism comprises a means for validating the compliance mechanism on the client system, a means for preventing decryption of the media content on the client system, and means for providing kernel level extension information to the framework. In addition, if a portion of the compliance mechanism is invalidated, there is a means for disabling the output of the media content on the client system having the Macintosh

20 operating system operating thereon.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the
5 description, serve to explain the principles of the invention.

FIGURE 1 is a block diagram of an exemplary computer system that can be utilized in accordance with an embodiment of the present invention.

10 FIGURE 2 is a block diagram of an exemplary network environment that can be utilized in accordance with an embodiment of the present invention.

FIGURE 3 is a block diagram of a copyright compliance mechanism in accordance with an embodiment of the present invention.

15 FIGURE 4 is an exemplary system for implementing a copyright compliance mechanism in accordance with an embodiment of the present invention.

FIGURE 5A is a data flow block diagram showing an implementation of a copyright
20 compliance mechanism for preventing unauthorized recording of media files, in accordance with one embodiment of the present invention.

FIGURE 5B is a data flow block diagram showing an implementation of a component of a copyright compliance mechanism for preventing unauthorized recording
25 of media files, in accordance with another embodiment of the present invention.

FIGURE 5C is a data flow block diagram showing an implementation of a copyright compliance mechanism for preventing unauthorized output of media files, in accordance with one embodiment of the present invention.

5

FIGURE 5D is a data flow block diagram showing an implementation of a copyright compliance mechanism for preventing unauthorized output of media files through media file capture at a kernel level, in accordance with one embodiment of the present invention.

10 FIGURE 6 is a block diagram of an environment for preventing unauthorized copying of a media file, in accordance with one embodiment of the present invention.

FIGURES 7A, 7B, and 7C are a flowchart of steps performed in accordance with an embodiment of the present invention for providing a copyright compliance mechanism to a
15 network of client and server computer systems.

FIGURE 8 is a diagram of an exemplary global media delivery system in which a copyright compliance mechanism can be implemented in accordance with an embodiment of the present invention.

20

FIGURE 9 is a block diagram of a copyright compliance mechanism installable from a media storage device, in accordance with one embodiment of the present invention.

FIGURE 10 is a block diagram of a communicative environment for controlling unauthorized reproduction of protected media files disposed on a media storage device, in accordance with one embodiment of the present invention.

5

FIGURE 11 is a data flow block diagram showing an implementation of a copyright compliance mechanism for preventing unauthorized reproduction of a protected media file located on a media storage device, in accordance with one embodiment of the present invention.

10

FIGURE 12 is a block diagram of components of a usage compliance mechanism installable from a media storage device upon which protected media files are disposed, in accordance with one embodiment of the present invention.

15 FIGURE 13 is a block diagram of components of a usage compliance mechanism and content disposed on a media storage device, in accordance with one embodiment of the present invention.

FIGURE 14 is a block diagram of a communicative environment for controlling
20 presentation of content on a media storage device, in accordance with one embodiment of the present invention.

FIGURE 15 is a data flow block diagram showing an implementation of a usage
compliance mechanism for controlling presentation of content disposed on a media storage
25 device, in accordance with one embodiment of the present invention.

FIGURE 16 is a flowchart of a process for controlling presentation of content disposed on a media storage device, in accordance with one embodiment of the present invention.

5

FIGURE 17 is a block diagram of a network environment for sharing media content among nodes within a network in accordance with one embodiment of the present invention.

10

FIGURE 18 is a block diagram of components within an exemplary usage compliance mechanism configured for utilization in a distributed network topology for controlling media sharing among nodes in a network, in accordance with one embodiment of the present invention.

15

FIGURE 19 is an illustrated data flow of an exemplary system for controlling media sharing among multiple nodes communicatively coupled in a network in accordance with one embodiment of the present invention.

20

FIGURE 20 is a data flow block diagram showing an implementation of a copyright compliance mechanism for preventing unauthorized recording of media files on a Macintosh operating system, in accordance with one embodiment of the present invention.

25

FIGURE 21 is a flowchart of a process for preventing unauthorized recording of electronic media on a Macintosh® operating system, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

Reference will now be made in detail to embodiments of the invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications, and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, to one of ordinary skill in the art, the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

Some portions of the detailed description which follows are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computing system or digital memory system. These descriptions and representations are the means used by those skilled in the data processing art to most effectively convey the substance of their work to others skilled in the art. A procedure, logic block, process, etc., is herein, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those involving physical manipulations of physical quantities. Usually, though not necessarily, these physical manipulations take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computing system or

similar electronic computing device. For reasons of convenience, and with reference to common usage, these signals are referred to as bits, values, elements, symbols, characters, terms, numbers, or the like, with reference to the present invention.

5 It should be borne in mind, however, that all of these terms are to be interpreted as referencing physical manipulations and quantities and are merely convenient labels and are to be interpreted further in view of terms commonly used in the art. Unless specifically stated otherwise as apparent from the following discussions, it is understood that discussions of the present invention refer to actions and processes of a computing system,
10 or similar electronic computing device that manipulates and transforms data. The data is represented as physical (electronic) quantities within the computing system's registers and memories and is transformed into other data similarly represented as physical quantities within the computing system's memories or registers, or other such information storage, transmission, or display devices.

15 In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. To one skilled in the art, the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in
20 order to avoid obscuring the present invention.

 Embodiments of the present invention are discussed primarily in the context of a network of computer systems such as a network of desktop, workstation, laptop, handheld, and/or other portable electronic device. For purposes of the present
25 application, the term "portable electronic device" is not intended to be limited solely to

conventional handheld or portable computers. Instead, the term “portable electronic device” is also intended to include many mobile electronic devices. Such mobile devices include, but are not limited to, portable CD players, MP3 players, mobile phones, portable recording devices, satellite radios, portable video playback devices (digital projectors), personal video eyewear, and other personal digital devices. Additionally, embodiments of the present invention are also well suited for implementation with theater presentation systems for public and/or private presentation in theaters, auditoriums, convention centers, etc.

Figure 1 is a block diagram illustrating an exemplary computer system 100 that can be used in accordance with embodiments of the present invention. It is noted that computer system 100 can be nearly any type of computing system or electronic computing device including, but not limited to, a server computer, a desktop computer, a laptop computer, or other portable electronic device. Within the context of embodiments of the present invention, certain discussed processes, procedures, and operations can be realized as a series of instructions (e.g., a software program) that reside within computer system memory units of computer system 100 and are executed by a processor(s) of computer system 100. When executed, the instructions cause computer system 100 to perform specific actions and exhibit specific behavior which is described in detail herein.

Computer system 100 of Figure 1 comprises an address/data bus 110 for communicating information, one or more central processors 101 coupled to bus 110 for processing information and instructions. Central processor(s) 101 can be a microprocessor or any alternative type of processor. Computer system 100 also

includes a computer usable volatile memory 102, e.g., random access memory (RAM), static RAM (SRAM), dynamic RAM (DRAM), synchronous dynamic RAM (SDRAM), double data rate RAM (DDR RAM), etc., coupled to bus 110 for storing information and instructions for processor(s) 101. Computer system 100 further includes a computer
5 usable non-volatile memory 103, e.g., read only memory (ROM), programmable ROM (PROM), electronically programmable ROM (EPROM), electrically erasable PROM (EEPROM), flash memory (a type of EEPROM), etc., coupled to bus 110 for storing static information and instructions for processor(s) 101. In one embodiment, non-volatile memory 103 can be removable.

10

System 100 also includes one or more signal generating and receiving devices (e.g., signal input/output device(s) 104) coupled to bus 110 for enabling computer 100 to interface with other electronic devices. Communication interface 104 can include wired and/or wireless communication functionality. For example, in one embodiment,
15 communication interface 104 is a serial communication port, but can alternatively be one of a number of well known communication standards and protocols, e.g., a parallel port, an Ethernet adapter, a FireWire (IEEE 1394) interface, a Universal Serial Bus (USB), a small computer system interface (SCSI), an infrared (IR) communication port, a Bluetooth wireless communication adapter, a broadband connection, a satellite link, an
20 Internet feed, a cable modem, and the like. In another embodiment, a digital subscriber line (DSL) can be implemented as signal input/output device 104. In such an instance, communication interface 104 may include a DSL modem.

Computer 100 of Figure 1 can also include one or more computer usable data
25 storage device(s) 108 coupled to bus 110 for storing instructions and information, in one

embodiment of the present invention. In one embodiment, data storage device 108 can be a magnetic storage device, e.g., a hard disk drive, a floppy disk drive, a zip drive, or other magnetic storage device. In another embodiment, data storage device 108 can be an optical storage device, e.g., a CD (compact disc), a DVD (digital versatile disc), or other alternative optical storage device. Alternatively, any combination of magnetic, optical, and alternative storage devices can be implemented, e.g., a RAID (random array of independent disks or random array of inexpensive discs) configuration. It is noted that data storage device 108 can be located internal and/or external of system 100 and communicatively coupled with system 100 utilizing wired and/or wireless communication technology, thereby providing expanded storage and functionality to system 100. It is further noted that nearly any portable electronic device, e.g., device 100a, can also be communicatively coupled with system 100 via utilization of wired and/or wireless communication technology, thereby expanding the functionality of system 100.

System 100 can also include an optional display device 105 coupled to bus 110 for displaying video, graphics, and/or alphanumeric characters. It is noted that display device 105 can be a CRT (cathode ray tube), a thin CRT (TCRT), a liquid crystal display (LCD), a plasma display, a field emission display (FED), video eyewear, a projection device (e.g., an LCD (liquid crystal display) or DLP (digital light projector), a movie theater projection system, and the like), or any other display device suitable for displaying video, graphics, and alphanumeric characters recognizable to a user.

Computer system 100 of Figure 1 further includes an optional alphanumeric input device 106 coupled to bus 110 for communicating information and command

selections to processor(s) 101, in one embodiment. Alphanumeric input device 106 includes alphanumeric and function keys. Computer 100 can also include an optional cursor control device 107 coupled to bus 110 for communicating user input information and command selections to processor(s) 101. Cursor control device 107 can be
5 implemented using a number of well known devices such as a mouse, a trackball, a track pad, a joy stick, a optical tracking device, a touch screen, etc. It is noted that a cursor can be directed and/or activated via input from alphanumeric input device 106 using special keys and key sequence commands. It is further noted that directing and/or activating the cursor can be accomplished by alternative means, e.g., voice
10 activated commands, provided computer system 100 is configured with such functionality.

Figure 2 is a block diagram of an exemplary network 200 in which embodiments of the present invention may be implemented. In one embodiment, network 200
15 enables one or more authorized client computer systems (e.g., 210, 220, and 230), each of which are coupled to Internet 201, to receive media content from a media content server 251 via the Internet 201 while preventing unauthorized client computer systems from accessing media stored in a database of content server 251.

20 Network 200 includes a web server 250 and content server 251 which are communicatively coupled to Internet 201. Further, web server 250 and content server 251 can be communicatively coupled without utilizing Internet 201, as shown. Web server 250, content server 251, and client computers 210, 220, and 230 can communicate with each other. It is noted that computers and servers of network 200 are well suited
25 to be communicatively coupled in various implementations. For example, web server

250, content server 251, and client computer systems 210, 220, and 230 of network 200 can be communicatively coupled via wired communication technology (e.g., twisted pair cabling, fiber optics, coaxial cable, etc.), or wireless communication technology, or a combination of wired and wireless communication technology.

5

Still referring to Figure 2, it is noted that web server 250, content server 251, and client computer systems 210, 220 and 230 can, in one embodiment, be each implemented in a manner similar to computer system 100 of Figure 1. However, the server and computer systems in network 200 are not limited to such implementation.

10 Additionally, web server 250 and content server 251 can perform various functionalities within network 200. It is also noted that, in one embodiment, web server 250 and content server 251 can both be disposed on a single or a plurality of physical computer systems.

15 Further, it is noted that network 200 can operate with and deliver any type of media content (e.g., audio, video, multimedia, graphics, information, data, software programs, etc.) in any format. In one embodiment, content server 251 can provide audio and video files to client computers 210-230 via Internet 201.

20 Figure 3 is a block diagram of an exemplary copyright compliance mechanism (CCM) 300, for controlling distribution of, access to, and/or copyright compliance of media files, in accordance with an embodiment of the present invention. In one embodiment, CCM 300 contains one or more software components and instructions for enabling compliance with DMCA (digital millennium copyright act) restrictions and/or
25 RIAA (recording industry association of America) licensing agreements regarding

media files. Additionally, CCM 300's software components and instructions further enable compliance with international recording restrictions such as those defined by the IFPI (international federation of phonographic industry), the ISRC (international standard recording industry), other foreign or international recording associations, and/or foreign or international licensing restrictions. In one embodiment, CCM 300 may be integrated into existing and/or newly developed media player and recorder applications. In another embodiment, CCM 300 may be implemented as a stand-alone mechanism but in conjunction with existing media player/recorder applications, such that CCM 300 is communicatively coupled to existing media player/recorder applications. Alternatively, CCM 300 can be installed as a stand-alone mechanism within a client computer system 210. Additionally, CCM 300 can be installed as a stand alone mechanism and/or as part of a bundled application from a media storage device, e.g., a CD, a DVD, an SD (secure digital card), and/or as part of an installation package. In another embodiment, CCM 300 can be installed in conjunction with a presentation of desired media content, e.g., listening to an audio file on a music CD, reading a document, viewing a video, etc. It is noted that, in one embodiment, CCM 300 may be installed on client system 210 in a clandestine manner, relative to a user.

There are currently two types of copyright licenses recognized by the digital millennium copyright act (DMCA) for the protection of broadcast copyrighted material. One of the broadcast copyright licenses is a compulsory license, also referred to as a statutory license. A statutory license is defined as a non-interactive license, meaning the user cannot select the song. Further, a caveat of this type of broadcast license is that a user must not be able to select a particular music file for the purpose of recording it to the user's computer system or other storage device. Another caveat of a

statutory license is that a media file is not available more than once for a given period of time. In one example, the period of time can be three hours.

The other type of broadcast license recognized by the DMCA is an interactive
5 licensing agreement. An interactive licensing agreement is commonly with the
copyright holder (e.g., a record company, the artist, etc.,) wherein the copyright holder
grants permission for a server, (e.g., web server 250 and/or content server 251) to
broadcast copyrighted material. Under an interactive licensing agreement, there are a
variety of ways that copyrighted material (e.g., music files) can be broadcast. For
10 example, one manner in which music files can be broadcast is to allow the user to select
and listen to a particular sound recording, but without the user enabled to make a
sound recording. This is commonly referred to as an interactive with “no save” license,
meaning that the end user is unable to save or store the media content file in a relatively
permanent manner. Additionally, another manner in which music files can be
15 broadcast is to allow a user to not only select and listen to a particular music file, but
additionally allow the user to save that particular music file to disc and/or burn the
music file to a CD, MP3 player, or other portable electronic device. This is commonly
referred to as an interactive with “save” license, meaning that the end user is enabled to
save, store, or burn to CD, the media content file.

20

It is noted that the DMCA allows for the “perfect” reproduction of the sound
recording. A perfect copy of a sound recording is a one-to-one mapping of the original
sound recording into a digitized form, such that the perfect copy is virtually
indistinguishable and/or has no audible differences from the original recording.

25

In one embodiment, CCM (copyright compliance mechanism) 300 can be stored in web server 250 and/or content server 251 of network 200 and is configured to be installed into each client computer system, e.g., 210, 220 and 230, enabled to access the media files stored within content server 251 and/or web server 250. Alternatively, copyright compliance mechanism 300 can be externally disposed and communicatively coupled with a client computer system 210 via, e.g., a portable media device 100a of Figure 1. In yet another embodiment, CCM 300 can be configured to be operable from a media storage device (e.g., 108) upon which media files may be disposed.

Copyright compliance mechanism 300 is configured to be operable while having portions of components, entire components, combinations of components, disposed within one or more memory units and/or data storage devices of a computer system, e.g., 210, 220, and/or 230.

Additionally, CCM 300 can be readily updated, (e.g., via Internet 201), to reflect changes or developments in the DMCA, copyright restrictions and/or licensing agreements pertaining to any media file, changes in current media player applications and/or the development of new media player applications, or to counteract subversive and/or hacker-like attempts to unlawfully obtain one or more media files. It is noted that updating CCM 300 can include, but is not limited to, updating portions of components, entire components and/or combinations of components of CCM 300.

Referring to Figure 3, CCM 300 can include instructions 301 for enabling client computer system 210 to interact with web server 250 and content server 251 of network

200. Instructions 301 enable client computer system 210 to interact with servers, (e.g., 250 and 251) in a network, (e.g., 200).

The copyright compliance mechanism 300 also includes, in one embodiment, a user ID generator 302, for generating a user ID or user key, and one or more cookie(s) which contain(s) information specific to the user and the user's computer system, e.g., 210. In one embodiment, the user ID and the cookie(s) are installed in computer system 210 prior to installation of the remaining components of the CCM 300. It is noted that the presence of a valid cookie(s) and a valid user ID/user key are verified by web server 250 before the remaining components of a CCM 300 can be installed, within one embodiment of the present invention. Additionally, the user ID/user key can contain, but is not limited to, the user's name, the user's address, the user's credit card number, an online payment account number, a verified email address, and an identity (username) and password selected by the user.

Furthermore, the cookie can contain, but is not limited to, information specific to the user, information regarding the user's computer system 210, (e.g., types of media applications operational therewithin), a unique identifier associated with computer system 210, e.g., a MAC (media access control) address, an IP address, and/or the serial number of the central processing unit (CPU) operable on computer system 210 and other information specific to the computer system and its user.

Additionally, in another embodiment, user biometrics may be combined with computer system 210 data and user data and incorporated into the generation of a user ID. Alternatively, biometric data may be used in a stand-alone implementation in the

generation of the user ID. Types of biometric data that may be utilized to provide a user ID and/or authorization may include, but is not limited to, fingerprint data, retinal scan data, handprint data, facial recognition data, and the like.

5 It is noted that the information regarding the client computer system, e.g., 210, the user of system 210, and an access key described herein can be collectively referred to as authorization data.

Advantageously, with information regarding the user and the user's computer
10 system, e.g., 210, web server 250 can determine when a user of one computer system, e.g., 210, has given their username and password to another user using another computer system, e.g., 220. Because the username, password, and the user's computer system 210 are closely associated, web server 250 can prevent unauthorized access to copyrighted media content, in one embodiment. It is noted that if web server 250
15 detects unauthorized sharing of usernames and passwords, it can block the user of computer system 210, as well as other users who unlawfully obtained the username and password, from future access to copyrighted media content available through web server 250. Web server 250 can invoke blocking for any specified period of time, e.g., for a matter of minutes, hours, months, years, or longer, or permanently.

20

Still referring to Figure 3, copyright compliance mechanism 300 further includes a coder/decoder (codec) 303 that, in one embodiment, is adapted to perform, but is not limited to, encoding/decoding of media files, compressing/decompressing of media files, and detecting that delivered media files are encrypted as prescribed by CCM 300.
25 In the present embodiment, coder/decoder 303 can also extract key fields from a header

attached to each media content file for, in part, verification that the file originated from a content server, e.g., 251. It is noted that CCM 300 can include one or more codecs similar to codec 303.

5 In the present embodiment, coder/decoder 303 can also perform a periodic and repeated check of the media file, while the media file is passed to the media player application, (e.g., in a frame by frame basis or in a buffer by buffer basis), to ensure that CCM 300 rules are being enforced at any particular moment during media playback. It is noted that differing coder/decoders 303 can be utilized in conjunction with various
10 types of copyrighted media content including, but not limited to, audio files, video files, graphical files, alphanumeric files and the like, such that any type of media content file can be protected in accordance with embodiments of the present invention.

 Within Figure 3, copyright compliance mechanism 300 also includes one or more
15 agent programs 304 which are configured to engage in dialogs and negotiate and coordinate transfer of information between a computer system, (e.g., 210, 220, or 230), a server, (e.g., web server 250 and/or content server 251), and/or media player applications, with or without recording functionality, that are operable within a client computer system, in one embodiment. In the present embodiment, agent program 304
20 can also be configured to maintain system state, verify that other components are being utilized simultaneously, to be autonomously functional without knowledge of the client, and can also present messages, (e.g., error messages, media information, advertising, etc.), via a display window or electronic mail. This enables detection of proper skin implementation and detection of those applications that are running. It is

noted that agent programs are well known in the art and can be implemented in a variety of ways in accordance with the present embodiment.

Copyright compliance mechanism 300 also includes one or more system hooks 305, in one embodiment of the present invention. A system hook 305 is, in one embodiment, a library that is installed in a computer system (e.g., 210) that intercepts system wide events. For example, a system hook 305, in conjunction with skins 306, can govern certain properties and/or functionalities of media player applications operating within the client computer system, e.g., 210, including, but not limited to, mouse click shortcuts, keyboard shortcuts, standard system accelerators, progress bars, save functions, pause functions, rewind functions, skip track functions, forward track preview, copying to CD, copying to a portable electronic device, and the like.

It is noted that the term govern or governing, for purposes of the present invention, can refer to a disabling, deactivating, enabling, activating, etc., of a property or function. Governing can also refer to an exclusion of that function or property, such that a function or property may be operable but unable to perform in the manner originally intended. For example, during the playing of a media file, the progress bar may be selected and moved from one location on the progress line to another without having an effect on the play of the media file.

Within Figure 3, it is further noted that codec 303 compares the information for the media player application operating on client computer system, e.g., 210, with a list of "signatures" associated with known media recording applications. In one embodiment, the signature can be, but is not limited to, a unique identifier of a media

player application which can consist of the window class of the application along with a product name string which is part of the window title for the application.

Advantageously, when new media player applications are developed, their signatures can be readily added to the signature list via an update of CCM 300 described herein.

5

The following C++ source code is an exemplary implementation of the portion of a codec 303 for performing media player application detection, in accordance with an embodiment of the present invention. In another embodiment, the following source code can be modified to detect kernel streaming mechanisms operable within a client system, (e.g., 210).

```
int
IsRecorderPresent(TCHAR *    szAppClass,
                  TCHAR *    szProdName)
15 {
    TCHAR    szWndText[_MAX_PATH];    /* buffer to receive title string for
window */
    HWND     hWnd;                    /* handle to target window for operation */
    int      nRetVal;    /* return value for operation */
20
    /* initialize variables */
    nRetVal = 0;

    if ( _tcscmp(szAppClass, _T("#32770"))
25     == 0)
    {
        /* attempt to locate dialog box with specified window title */
        if ( FindWindow((TCHAR *) 32770, szProdName)
            != (HWND) 0)
30        {
            /* indicate application found */
            nRetVal = 1;
        }
    }
35     else
    {
        /* attempt to locate window with specified class */
        if ( (hWnd = FindWindow(szAppClass, (LPCTSTR) 0))
```

```

        != (HWND) 0)
    {
        /* attempt to retrieve title string for window */
        if ( GetWindowText(hWnd,
5              szWndText,
              _MAX_PATH)
            != 0)
        {
            /* attempt to locate product name within title string */
            if ( _tcsstr(szWndText, szProdName)
10              != (TCHAR *) 0)
            {
                /* indicate application found */
                nRetVal = 1;
15            }
        }
    }

    /* return to caller */
    return nRetVal;
20 }

```

Within Figure 3, it is further noted that codec 303 can also selectively suppress
 25 waveform input/output operations to prevent recording of copyrighted media on a
 client computer system, (e.g., 210). For example, codec 303, subsequent to detection of
 bundled media player applications operational in a client computer system (e.g., 210)
 can stop or disrupt the playing of a media content file. This can be accomplished, in one
 embodiment, by redirecting and/or diverting certain data pathways that are commonly
 30 used for recording, such that the utilized data pathway is governed by the copyright
 compliance mechanism 300. In one embodiment, this can be performed within a driver
 shim, (e.g., wave driver shim 309 of Figures 5A, 5B, 5C , and 5D.

A driver shim can be utilized for nearly any software output device, such as a
 35 standard Windows™ waveform output device (e.g., Windows™ Media Player), or a
 hardware output device (e.g., speakers or headphones). Client computer system 210 is

configured such that the driver shim (e.g., 309) appears as the default waveform media device to client level application programs. Thus, requests for processing of waveform media input and/or output will pass through the driver shim prior to being forwarded to the actual waveform audio driver, (e.g., media device driver 505 of Figures 5A-5D).

5 Such waveform input/output suppression can be triggered by other components (e.g., agent 304) of CCM 300 to be active when a recording operation is initiated by a client computer system (e.g., 210) during the play back of media files which are subject to the DMCA.

10 It is noted that alternative driver shims can be implemented for nearly any waveform output device including, but not limited to, a Windows™ Media Player. It is further noted that the driver shim can be implemented for nearly any media in nearly any format including, but not limited to, audio media files, audio input and output devices, video, graphic and/or alphanumeric media files and video input and output
15 devices.

The following C++ source code is an exemplary implementation of a portion of a codec 303 and/or a custom media device driver 307 for diverting and/or redirecting certain data pathways that are commonly used for recording of media content, in
20 accordance with an embodiment of the present invention.

```
DWORD
_stdcall
widMessage(UINT      uDevId,
25      UINT      uMsg,
      DWORD      dwUser,
      DWORD      dwParam1,
      DWORD      dwParam2)
```

```

{
    BOOL      bSkip;      /* flag indicating operation to be skipped */
    HWND      hWndMon;    /* handle to main window for monitor */
    DWORD      dwRetVal;   /* return value for operation */
5
    /* initialize variables */
    bSkip = FALSE;
    dwRetVal = (DWORD) MMSYSERR_NOTSUPPORTED;

10
    if (uMsg == WIDM_START)
    {
        /* attempt to locate window for monitor application */
        if ( (hWndMon = FindMonitorWindow())
            != (HWND) 0)
15
        {
            /* obtain setting for driver */
            bDrvEnabled = ( SendMessage(hWndMon,
                                     uiRegMsg,
                                     0,
                                     0)
20
                               == 0)
                               ? FALSE : TRUE;

        }

25
        if (bDrvEnabled == TRUE)
        {
            /* indicate error in operation */
            dwRetVal = MMSYSERR_NOMEM;

30
            /* indicate operation to be skipped */
            bSkip = TRUE;
        }
    }

35
    if (bSkip == FALSE)
    {
        /* invoke entry point for original driver */
        dwRetVal = CallWidMessage(uDevId, uMsg, dwUser, dwParam1,
40
        dwParam2);
    }

    /* return to caller */
    return dwRetVal;
}
45

```

It is noted that when properly configured, system hook 305 can govern nearly any function or property within nearly any media player application that may be

operational within a client computer system, (e.g., 210). In one embodiment, system hook 305 is a DLL (dynamic link library) file. It is further noted that system hooks are well known in the art, and are a standard facility in a Microsoft Windows™ operating environment, and accordingly can be implemented in a variety of ways. However, it is also noted that system hook 305 can be readily adapted for implementation in alternative operating systems, e.g., Apple™ operating systems, Sun Solaris™ operating systems, Linux operating systems, and nearly any other operating system.

In Figure 3, copyright compliance mechanism 300 also includes one or more skins 306, which can be designed to be installed in a client computer system, (e.g., 210, 220, and 230). In one embodiment, skins 306 are utilized to assist in client side compliance with the DMCA (digital millennium copyright act) regarding copyrighted media content. Skins 306 are customizable interfaces that, in one embodiment, are displayed on a display device (e.g., 105) of computer system 210 and provide functionalities for user interaction of delivered media content. Additionally, skins 306 can also provide a display of information relative to the media content file including, but not limited to, song title, artist name, album title, artist biography, and other features such as purchase inquiries, advertising, and the like.

Furthermore, when system hook 305 is unable to govern a function of the media player application operable on a client computer system (e.g., 210) such that client computer system could be in non-compliance with DMCA and/or RIAA restrictions, a skin 306 can be implemented to provide compliance.

Differing skins 306 can be implemented depending upon the restrictions applicable (e.g., DMCA and/or RIAA) to each media content file. For example, in one embodiment, a skin 306a may be configured for utilization with a media content file protected under a non-interactive agreement (DMCA), such that skin 306a may not
5 include a pause function, a stop function, a selector function, and/or a save function, etc. Another skin, e.g., skin 306b may, in one embodiment, be configured to be utilized with a media content file protected under an interactive with "no save" agreement (DMCA), such that skin 306b may include a pause function, a stop function, a selector function, and for those media files having an interactive with "save" agreement, a save
10 or a burn to CD function.

Still referring to Figure 3, it is further noted that in the present embodiment, each skin 306 can have a unique name and signature. In one embodiment, skin 306 can be implemented, in part, through the utilization of an MD (message digest) 5 hash table or
15 similar algorithm. An MD5 hash table can, in one implementation, be a check-sum algorithm. It is well known in the art that a skin, e.g., skin 306, can be renamed and/or modified to incorporate additional features and/or functionalities in an unauthorized manner. Since modification of the skin would change the check sum and/or MD5 hash, without knowledge of the MD5 hash table, changing the name or modification of the
20 skin may simply serve to disable the skin, in accordance with one embodiment of the present invention. Since copyright compliance mechanism (CCM) 300 verifies skin 306, MD5 hash tables advantageously provide a deterrent against modifications made to the skin 306.

In one embodiment, CCM 300 also includes one or more custom media device driver(s) 307 for providing an even greater measure of control over the media stream while increasing compliance reliability. A client computer system (e.g., 210) can be configured to utilize a custom media device application (e.g., custom media device 310 of Figures 5B, 5C, and 5D) to control unauthorized recording of media content files. A custom media device application can be, but is not limited to, a custom media audio device application for media files having sound content, a custom video device application for media files having graphical and/or alphanumeric content, etc. In one embodiment, custom media device 310 of Figure 5B is an emulation of the custom media device driver 307. With reference to audio media, the emulation is performed in a waveform audio driver associated with custom media device 310. Driver 307 is configured to receive a media file being outputted by system 210 prior to the media file being sent to a media output device (e.g., media output device 570) and/or a media output application (e.g., recording application 502). Examples of a media output device includes, but is not limited to, a video card for video files, a sound card for audio files, etc. Examples of a recording application can include, but is not limited to, CD burner applications for writing to another CD, ripper applications which capture the media file and change the format of the media file, e.g., from a CD audio file to an .mpeg audio file, and/or a .wav file, and/or an ogg vorbis file, and various other media formats. In one embodiment, client computer system 210 is configured with a custom media device driver 307 emulating custom media device 310, and which is system 210's default device driver for media file output. In one embodiment, an existing GUI (graphical user interface) can be utilized or a GUI can be provided, e.g., by utilization of skin 306 or a custom web based player application or as part of a CCM 300 installation bundle, for forcing or requiring system 210 to have driver 307 as the default driver.

Therefore, when a media content file is received by system 210 from server 251, the media content file is playable, provided the media content file passes through the custom media device application (e.g., 310 of Figure 5B), emulated by custom media device driver 307, prior to being outputted. However, if an alternative media player application is selected, delivered media files from server 251 will not play on system 210.

Thus, secured media player applications would issue a media request to the driver (e.g., 307) for the custom media device 310 which then performs necessary media input suppression, (e.g., waveform suppression for audio files), prior to forwarding the request to the default Windows™ media driver, (e.g., waveform audio driver for audio files).

Within Figure 3, it is noted that requests for non-restricted media files can pass directly through custom media device driver 307 to a Windows™ waveform audio driver operable on system 210, thus reducing instances of incompatibilities with existing media player applications that utilize waveform media, (e.g., audio, video, etc.). Additionally, media player applications that do not support secured media would be unaffected. It is further noted that for either secured media or non-restricted media, (e.g., audio media files), waveform input suppression can be triggered by other components of CCM 300, (e.g., agents 304, system hooks 305, and skins 306, or a combination thereof), to be active when a recording operation is initiated simultaneously with playback of secured media files, e.g., audio files. Custom device drivers are well known and can be coded and implemented in a variety of ways

including, but not limited to, those found at developers network web sites, (e.g., a Microsoft™ or alternative OS (operating system) developer web sites).

Advantageously, by virtue of system 210 being configured with a custom media device as the default device driver, (e.g., 310 of Figures 5B, 5C, and 5D), that is an emulation of a custom media device driver 307, those media player applications that require their particular device driver to be the default driver, e.g., Total Recorder, etc., are rendered non-functional for secured media. Further advantageous is that an emulated custom media device provides no native support for those media player applications used as a recording mechanism, e.g., DirectSound capture, (direct sound 504 of Figures 5A, 5B, 5C, and 5D) etc., that are able to bypass user-mode drivers for most media devices. Additionally, by virtue of the media content being sent through device driver 307, thus effectively disabling unauthorized saving/recording of media files, in one embodiment, media files that are delivered in a secured delivery system do not have to be encrypted, although, in another embodiment, they still may be encrypted. By virtue of non-encrypted media files utilizing less storage space and network resources than encrypted media files, networks having limited resources can utilize the functionalities of driver 307 of CCM 300 to provide compliance with copyright restrictions and/or licensing agreements applicable with a media content file without having the processing overhead of encrypted media files.

Figure 4 is an exemplary system 400 for implementing a copyright compliance mechanism in accordance with an embodiment of the present invention. Specifically, system 400 illustrates web server 250, content server 251, or a combination of web server 250 and content server 251 installing a copyright compliance mechanism (e.g.,

300) in a client's computer system (e.g., 210) for controlling media file distribution and controlling user access and interaction of copyrighted media files, in one embodiment of the present invention.

5 Client computer system 210 can communicatively couple with a network (e.g., 200) to request a media file, a list of available media files, or a play list of audio files, e.g., MP3 files, etc. In response, web server 250 determines if the request originates from a registered user authorized to receive media files associated with the request. If the user is not registered with the network, web server 250 can initiate a registration
10 process with the requesting client 210. Client registration can be accomplished in a variety of ways. For example, web server 250 may deliver to client 210 a registration form having various text entry fields into which the user can enter required information. A variety of information can be requested from the user by web server 250 including, but not limited to, user's name, address, phone number, credit card number,
15 online payment account number, biometric identification (e.g., fingerprint, retinal scan, etc.), verifiable email address, and the like. In addition, registration can, in one embodiment, include the user selecting a username and password.

 Still referring to Figure 4, web server 250 can, in one embodiment, detect
20 information related to the client's computer system 210 and store that information in a user/media database 450. For example, web server 250 can detect a unique identifier of client computer system 210. In one embodiment, the unique identifier can be the MAC (media access control) address of a NIC (network interface card) of client computer system 210 or the MAC address of the network interface adapter integrated on the
25 motherboard of system 210. It is understood that a NIC enables a client computer

system 210 to access web server 250 via a network such as Internet 201. It is well known that each NIC typically has a unique identifying number MAC address. Further, web server 250 can, in one embodiment, detect and store (also in database 450) information regarding the type(s) of media player application(s), e.g., Windows Media Player™, Real Player™, iTunes player™ (Apple), Live 365™ player, and those media player applications having recording functionality (e.g., Total Recorder, Cool Edit 2000, Sound Forge, Sound Recorder, Super MP3 Recorder, and the like) that are present and operable in client computer system 210. In one embodiment, the client information is verified for accuracy and is then stored in a user database (e.g., 450) within web server 250.

Subsequent to registration completion, creation of the user ID and password, and obtaining information regarding client computer system 210, all or part of this information can be installed in client computer system 210. In one embodiment, client computer system 210 information can be in the form of a cookie. Web server 250 then verifies that the user and client computer system 210 data is properly installed therein and that their integrity has not been compromised. Subsequently, web server 250 installs a copyright compliance mechanism (e.g., 300) into the client's computer system, e.g., 210, in one embodiment of the present invention. It is noted that web server 250 may not initiate installation of CCM 300 until the user ID, password, and client computer system 210 information is verified. A variety of common techniques can be employed to install an entire CCM 300, portions of its components, entire components, and/or combinations or a function of its components. For example, copyright compliance mechanism 300 can be installed in a hidden directory within client computer system 210, thereby preventing unauthorized access to it. In one embodiment

it is noted that unless CCM 300 is installed in client computer system 210, its user will not be able to request, access, or have delivered thereto, media files stored by web server 250 and/or content server 251.

5 Referring still to Figure 4, upon completion of client registration and installation of CCM 300, client computer system 210 can then request a media play list or a plurality of play lists, etc. In response, web server 250 determines whether the user of client computer system 210 is authorized to receive the media play list associated with the request. In one embodiment, web server 250 can request the user's username and
10 password. Alternatively, web server 250 can utilize user database 450 to verify that computer 210 is authorized to receive a media play list. If client computer 210 is not authorized, web server 250 can initiate client registration, as described herein. Additionally, web server 250 can disconnect computer 210 or redirect it to an alternative web site. Regardless, if the user and client computer system 210 are not authorized,
15 web server 250 will not provide the requested play list to client computer system 210.

However, if client computer system 210 is authorized, web server 210 can check copyright compliance mechanism 300 within data base 450 to determine if it, or any of the components therein, have been updated since the last time client computer system
20 210 logged into web server 250. If a component of CCM 300 has been updated, web server 250 can install the updated component and/or a more current version of CCM 300 into client computer system 210, e.g., via Internet 201. If CCM 300 has not been updated, web server 250 can then deliver the requested media play list to system 210 via Internet 201 along with an appended user key or user identification (ID). It is noted
25 that user database 450 can also include data for one or more media play lists that can be

utilized to provide a media play list to client computer system 210. Subsequently, the user of client computer system 210 can utilize the received media play list in combination with the media player application operating on system 210 to transmit a delivery request for one or more desired pieces of media content from web server 250.

5 It is noted that the delivery request contains the user key for validation purposes.

Still referring to Figure 4, upon receiving the media content delivery request, web server 250 can then check the validity of the requesting media application and the attached user key. In one embodiment, web server 250 can utilize user database 450 to
10 check their validity. If either or both are invalid, web server 250, in one embodiment, can redirect unauthorized client computer system 210 to an alternative destination to prevent abuse of the system. However, if both the requesting media application and the user key are valid, CCM 300 verifies that skins 306 are installed in client computer system 210. Additionally, CCM 300 further verifies that system hook(s) 305 have been
15 run or are running to govern certain functions of those media player applications operable within client computer system 210 that are known to provide non-compliance with one or more restricted use standards such as the DMCA and/or the RIAA. Additionally, CCM 300 further diverts and/or redirects certain pathways that are commonly used for recording, e.g., driver 307 of Figure 5A, device 310 of Figure 5B,
20 device 570 of Figure 5C, and driver 505 of Figure 5D. Once CCM 300 has performed the above described functions, web server 250 then, in one embodiment, issues to the client computer 210 a redirect command to the current address location of the desired media file content along with an optional time sensitive access key, e.g., for that hour, day, or other defined timeframe.

25

In response to the client computer system 210 receiving the redirect command from web server 250, the media player application operating on client computer system 210 automatically transmits a new request and the time sensitive access key to content server 251 for delivery of one or more desired pieces of media content. The validity of the time sensitive access key is checked by content server 251. If invalid, unauthorized client computer 210 is redirected by content server 250 to protect against abuse of the system and unauthorized access to content server 251. If the time sensitive access key is valid, content server 251 retrieves the desired media content from content database 451 and delivers it to client computer system 210. It is noted that, in one embodiment, the delivered media content can be stored in hidden directories and/or custom file systems that may be hidden within client computer system 210 thereby preventing future unauthorized distribution. In one embodiment, an HTTP (hypertext transfer protocol) file delivery system is used to deliver the requested media files, meaning that the media files are delivered in their entirety to client computer system 210, as compared to streaming media which delivers small portions of the media file.

Still referring to Figure 4, it is noted that each media file has had, in one embodiment, a header attached therewith prior to delivery of the media file. In one embodiment, the header can contain information relating to the media file, e.g., title or media ID, media data such as size, type of data, and the like. The header can also contain a sequence or key that is recognizable to copyright compliance mechanism 300 that identifies the media file as originating from content server 251. In one embodiment, the header sequence/key can also contain instructions for invoking the licensing agreements and/or copyright restrictions that are applicable to that particular media file.

Additionally, if licensing agreements and/or copyright restrictions are changed, developed, or created, or if new media player applications, with or without recording functionality, are developed, CCM 300 has appropriate modifications made to portions of components, entire components, combinations of components, and/or the entire CCM 300 to enable continued compliance with licensing agreements and/or copyright restrictions. Furthermore, subsequent to modification of copyright compliance mechanism 300, modified portions of, or the entire updated CCM 300 can be installed in client computer system 210 in a variety of ways. For example, the updated CCM 300 can be installed during client interaction with web server 250, during user log-in, and/or while client computer system 210 is receiving the keyed play list.

Referring still to Figure 4, it is further noted that, in one embodiment, the media files and attached headers can be encrypted prior to being stored within content server 251. In one embodiment, the media files can be encrypted utilizing randomly generated keys. Alternatively, variable length keys can be utilized for encryption. It is noted that the key to decrypt the encrypted media files can be stored in database 450, content database 451 or in some combination of databases 450 and 451. It is further noted that the messages being passed back and forth between client computer system 210 and web server 250 can also be encrypted, thereby protecting the media files and the data being exchanged from unauthorized use or access. There are a variety of encryption mechanisms and programs that can be implemented to encrypt this data including, but not limited to, exclusive OR, shifting with adds, public domain encryption programs such as Blowfish, and non-public domain encryption mechanisms. It is also noted that each media file can be uniquely encrypted, such that if the encryption code is cracked for one media file, it is not applicable to other media files. Alternatively, groups of

media files can be similarly encrypted. Furthermore, in another embodiment, the media files may not be encrypted when being delivered to a webcaster known to utilize a proprietary media player application, e.g., custom media device driver 307.

5 Subsequent to media file decryption, the media file may be passed through CCM 300, (e.g., coder/decoder 303), to a media player application operating on client computer system 210, (e.g. playback application 501 of Figures 5A, 5B, 5C, 5D, and 6A), which can then access and utilize the delivered high fidelity media content, enabling its user(s) to experience the media content, e.g., listen to it, watch it, view it, or the like. In
10 one embodiment of the present invention, a specialized or custom media player may or may not be required to experience the media content, (e.g., skin 306 of Figure 3). A skin 306 may be necessary when CCM 300 cannot modify an industry standard media player application to comply with copyright restrictions and/or licensing agreements in accordance with the DMCA. Alternatively, an industry standard media player can be
15 utilized by client computer system 210 to experience the media content. Typically, many media player applications are available and can include, but are not limited to, Windows™ Media Player™ for PCs (personal computers), iTunes™ Player or QuickTime™ for Apple computers, and XMMS player for computers utilizing a Linux operating system. Regardless of the media player application utilized, while the media
20 file is passed to the media player application, e.g., in a frame by frame basis or in a buffer, coder/decoder 303 will repeatedly ensure that CCM 300 rules are being enforced at any particular moment during media playback, shown as step 750 of Figure 7C.

 As the media file content is delivered to the media player application,
25 periodically, (e.g., after a specified number of frames, after a defined period of time, or

any desired time or data period), coder/decoder 303 repeatedly determines whether or not all the rules, as defined by CCM 300, are enforced. If the rules are not enforced, (e.g., a user opening up a recording application such as Total Recorder or an alternative application), the presentation of the media content is, in one embodiment, suspended or
5 halted. In another embodiment, the presentation of the media content can be modified to output the media content in a non-audible manner, (e.g., silence). In yet another embodiment, the media content may be audible but recording functionality can be disabled, such that the media content cannot be recorded. These presentation stoppages are collectively shown as step 751 of Figure 7C.

10

If the rules in accordance with CCM 300 are enforced, the codec/decoder 303 retrieves a subsequent portion of the media content that is stored locally in client computer system 210. The newly retrieved portion of the media file is then presented by the client's media player application. While the newly retrieved portion is
15 presented, CCM 300 again checks that the rules are enforced, and retrieves an additional portion of the media file or suspends presentation of the media file if the rules are not being enforced. These operations are performed repeatedly throughout the playback of the media file, in a loop environment, until the media file's contents have been presented in their entirety. Advantageously, by constantly monitoring
20 during playing of media files, CCM 300 can detect undesired activities and enforces those rules as defined by CCM 300.

Figure 5A is an exemplary logic/bit path block diagram 500A showing utilization of a wave shim driver, (e.g., 309 of Figure 3), in conjunction with copyright
25 compliance mechanism 300, for selectively controlling recording of copyrighted media

received by a client computer system, (e.g., system 210), in one embodiment of the present invention. Copyright compliance mechanism 300 is, in one embodiment, installed and operational on client system 210 in the manner described herein.

5 In one embodiment, a copyright compliance mechanism 300 is shown as being communicatively coupled with a media playback application 501 via coupling 520. Therefore, CCM 300 is enabled to communicate with playback application 501. In one embodiment, CCM 300 can be integrated into a media playback application. CCM 300 is also coupled to and controls a selectable switch 311 in wave shim driver 309 (as
10 described in Figure 3) via coupling 522. CCM 300 is further coupled to and controls a selectable switch 511 in direct sound 504 via coupling 521. Depending upon the copyright restrictions and licensing agreements applicable to an incoming media file, (e.g., 499), CCM 300 controls whether switches 311 and 511 are open (shown), thus preventing incoming media 499 from reaching a media recording application, or closed
15 (not shown) to allow recording of incoming media 499.

 For example, incoming media 499 may originate from a content server, e.g., 251, coupled to system 210. In another example, incoming media 499 may originate from a personal recording/electronic device, (e.g., a MP3 player/recorder or similar device),
20 coupled to system 210. Alternatively, incoming media 499 may originate from a magnetic, optical or alternative media storage device inserted into a media device player coupled to system 210, (e.g., a CD or DVD inserted into a CD or DVD player), a hard disk in a hot swappable hard drive, an SD (secure digital card) inserted into a SD reader, and the like. In yet another example, incoming media 499 may originate from
25 another media player application or media recording application. Incoming media 499

may also originate from a satellite radio feed (e.g., XM radio), a personal communication device (e.g., a mobile phone), a cable television radio input, e.g., DMX (digital music express), a digital distribution and/or a public presentation source via a network, Internet or other communication connection, pay-per-view and/or pay-per-play system, or a set-top box. It is noted that incoming media 499 can originate from nearly any source that can be coupled to system 210. However, regardless of the source of incoming media 499, embodiments of the present invention, described herein, can prevent unauthorized recording of the media 499.

Figure 5A shows a media playback application 501, (e.g., an audio, video, or other media player application), operable within system 210 and configured to receive incoming media 499. Playback application 501 can be a playback application provided by an operating system, (e.g., Media Player for Windows™ by Microsoft), a freely distributed playback application downloadable from the Internet, (e.g., RealPlayer or LiquidAudio), a playback application provided by a webcaster, (e.g., PressPlay), or a playback application commercially available.

Media device driver 505 in one embodiment, may be a software driver for a sound card coupled to system 210 having a media output device 570, (e.g., speakers or headphones), coupled therewith for media files having audio content. In another implementation, media device driver 505 may be a software driver for a video card coupled with a display device, (e.g., 105), for displaying media files having alphanumeric and/or graphical content, and so on. With reference to audio files, it is well known that a majority of recording applications assume a computer system, (e.g., 210), has a sound card disposed therein, providing full-duplex sound functionality to

system 210. This means media output driver 505 can simultaneously cause playback and recording of incoming media files 499. For example, media device driver 505 can playback media 499 along wave-out line 539 to media output device 570 (e.g., speakers for audible playback) via wave-out line 580 while outputting media 499 on wave-out
5 line 540 to eventually reach recording application 502.

For purposes of Figures 5A, 5B, 5C, and 5D, the terms wave-in line and wave-out line are referenced from the perspective of media device driver 505. Additionally, for the most part, wave-in lines are depicted downwardly and wave-out lines are depicted
10 upwardly in Figures 5A, 5B, 5C, and 5D.

Continuing with Figure 5A, playback application 501 is coupled with an operating system (O/S) multimedia subsystem 503 via wave-in line 531. O/S multimedia subsystem 503 is coupled to a wave shim driver 309 via wave-in line 533
15 and wave-out line 546. O/S multimedia subsystem 503 is also coupled to recording application 502 via wave-out line 548. Operating system (O/S) multimedia subsystem 503 can be any O/S multimedia subsystem, e.g., a Windows™ multimedia subsystem for system 210 operating under a Microsoft O/S, a QuickTime™ multimedia subsystem for system 210 operating under an Apple O/S, and the like. Playback application 501 is
20 also coupled with direct sound 504 via wave-in line 551.

Direct sound 504, in one embodiment, may represent access to a hardware acceleration feature in a standard audio device, enabling lower level access to components within media device driver 505. In another embodiment, direct sound 504
25 may represent a path that can be used by a recording application, (e.g., Total Recorder),

that can be further configured to bypass the default device driver, (e.g., media device driver 505), to capture incoming media 499 for recording. For example, direct sound 504 can be enabled to capture incoming media 499 via wave-in line 551 and unlawfully output media 499 to recording application 502 via wave-out line 568, as well as media 499 eventually going to media device driver 505, the standard default driver.

Still referring to Figure 5A, wave shim driver 309 is coupled with media device driver 505 via wave-in line 537 and wave-out line 542. Media device driver 505 is coupled with direct sound 504 via wave-in line 553 which is shown to converge with wave-in line 537 at media device driver 505. Media device driver 505 is also coupled with direct sound 504 via wave-out line 566.

Wave-out lines 542 and 566 are shown to diverge from wave-out line 540 at media device driver 505 into separate paths. Wave-out line 542 is coupled to wave shim driver 309 and wave-out line 566 is coupled to direct sound 504. When selectable switches 311 and 511 are open (shown), incoming media 499 cannot flow to recording application 502, thus preventing unauthorized recording of it.

For example, incoming media 499 is received at playback application 501. Playback application 501 activates and communicates to CCM 300 regarding copyright restrictions and/or licensing agreements applicable to incoming media 499. If recording restrictions apply to media 499, CCM 300 can, in one embodiment, open switches 311 and 511, thereby blocking access to recording application 502 to effectively prevent unauthorized recording of media 499. In one embodiment, CCM 300 can detect if system 210 is configured with direct sound 504 selected as the default driver to

capture incoming media 499, via wave-in line 551, or a recording application is detected and/or a hardware accelerator is active, such that wave driver shim 309 can be bypassed by direct sound 504. Upon detection, CCM 300 can control switch 511 such that the output path, wave-out line 568, to recording application 502 is blocked. It is further noted that CCM 300 can detect media recording applications and devices as described herein, with reference to Figure 3.

Alternatively, if media device driver 505 is selected as the default driver, incoming media 499 is output from playback application 501 to O/S multimedia subsystem 503 via wave-in line 531. From subsystem 503, media 499 is output to wave shim driver 309 via wave-in line 533. The wave shim driver 309 was described herein with reference to Figure 3. Media 499 is output from wave shim driver 309 to media device driver 505 via wave-in line 537. Once received by media device driver 505, media 499 can be output via wave-out line 539 to media output device 570 coupled therewith via wave-out line 580. Additionally, media device driver 505 can simultaneously output media 499 on wave-out line 540 back to wave shim driver 309. Dependent upon recording restrictions applicable to media 499, CCM 300 can, in one embodiment, close switch 311 (not shown as closed), thereby allowing media 499 to be output from wave shim driver 309 to subsystem 503 (via wave-out line 546) and then to recording application 502 via wave-out line 548. Alternatively, CCM 300 can also open switch 311, thereby preventing media 499 from reaching recording application 502.

It is noted that by virtue of CCM 300 controlling both switches 311 and 511, and therefore controlling wave-out line 548 and wave-out line 568 leading into recording application 502, incoming media files, (e.g., 499), can be prevented from being recorded

in an unauthorized manner in accordance with applicable copyright restrictions and/or licensing agreements related to the incoming media 499. It is also noted that embodiments of the present invention in no way interfere with or inhibit the playback of incoming media 499.

5

Figure 5B is an exemplary logic/bit path block diagram 500B of a client computer system, (e.g., 210), configured with a copyright compliance mechanism 300 for preventing unauthorized recording of copyrighted media according to an embodiment of the present invention. Copyright compliance mechanism 300 is, in one embodiment, coupled with and operational on client system 210 in the manner described herein with reference to Figures 4, 5A, 5C, 5D, 6, and 7.

Diagram 500B of Figure 5B is similar to diagram 500A of Figure 5A, with a few changes. Particularly, diagram 500B includes a custom media device 310 communicatively interposed between and coupled to O/S multimedia subsystem 503 and wave shim driver 309. Custom media device 310 is coupled to O/S multimedia subsystem via wave-in line 533 and wave-out line 546. Custom media device 310 is coupled with wave shim driver 309 via wave-in line 535 and wave-out line 544. Additionally, custom media device 310 is coupled with direct sound 504 via wave-in line 553 which converges with wave-in line 533 and wave-out line 566 which diverges from wave-out line 546, in one embodiment.

Diagram 500B also includes a media hardware output device 570 that is coupled to media device hardware driver 505 via line 580. Media hardware output device 570

can be, but is not limited to, a sound card for audio playback, a video card for video, graphical, alphanumeric output, and the like.

In one embodiment, CCM 300 is communicatively coupled with playback
5 application 501 via coupling 520, waveform driver shim 309 via coupling 522, and
custom media device 310, via coupling 525. CCM 300 is coupled to and controls
selectable switch 311 in waveform driver shim 309 via coupling 522. CCM 300 is also
coupled to and controls selectable switch 312 in custom audio device 310 via coupling
525. Depending upon the copyright restrictions and licensing agreements applicable to
10 an incoming media file, (e.g., media 499), CCM 300 controls whether switches 311 and
312 are open (shown), thus preventing the incoming media 499 from reaching a
recording application, or closed (not shown) so as to allow recording of the incoming
media 499.

15 Continuing with Figure 5B, direct sound 504 is coupled with custom media
device 310 via wave-in line 553, instead of being coupled with media device driver 505
(Figure 5A). In one embodiment, custom audio device 310 mandates explicit selection
through system 210, meaning that custom audio device 310 needs to be selected as a
default driver of system 210. By virtue of having the selection of custom media device
20 310 as the default driver of system 210, the data path necessary for direct sound 504 to
capture the media content can be selectively closed.

For example, incoming media 499 originating from nearly any source described
herein with reference to Figure 5A is received by media playback application 501 of
25 system 210. Playback application 501 communicates to CCM 300, via coupling 520, to

determine whether incoming media 499 is protected by any copyright restrictions and/or licensing agreements. Playback application 501 communicates with CCM 300 to control switch 311 and 312 accordingly. For example, if recording of incoming media 499 would violate applicable restrictions and/or agreements, switch 312 is in an open position (as shown), such that the output path to recording application 502, (e.g., wave-out line 548 and/or wave-out line 568), is effectively blocked thereby preventing unauthorized recording of media 499.

Alternatively, if media device driver 505 is selected as the default driver, incoming media 499 continues from O/S multimedia subsystem 503, through custom media device 310, wave driver shim 309, and into media device driver 505 where media 499 can be simultaneously output to media output device 570 via line 580, and output on wave-out line 540 wave shim driver 309 on wave-out line 542. However, by virtue of CCM 300 controlling switch 311, wave-out line 544 which eventually leads to recording application 502 is blocked, thus effectively preventing unauthorized recording of media 499.

It is noted that by virtue of CCM 300 controlling both switches 311 and 312 and therefore controlling wave-out line 548 and wave-out line 568, any incoming media files, (e.g., 499), can be prevented from being recorded in an unauthorized manner in accordance with applicable copyright restrictions and/or licensing agreements related to the incoming media 499.

Still referring to Figure 5B, it is further noted that custom media device 310 allows for unfettered playback of incoming media 499. Additionally, at any time during

playback of media 499, custom media device 310 can be dynamically activated by CCM 300.

Figure 5C is an exemplary logic/bit path block diagram 500C of a client computer system, (e.g., 210), configured with a copyright compliance mechanism 300 for preventing unauthorized output and unauthorized recording of copyrighted media according to an embodiment of the present invention. Copyright compliance mechanism 300 is, in one embodiment, coupled with and operational on client system 210 in the manner described herein with reference to Figures 4, 5A, 5B, 5D, 6, and 7.

Diagram 500C of Figure 5C is similar to diagram 500B of Figure 5B, with a few changes. Particularly, media hardware output device 570 is shown to include a switch 571 controlled by CCM 300 via communication line 523, similar to switches 311 and 312, for controlling output of incoming media 499. Diagram 500C includes media hardware output device 570 that is coupled with a media device driver 505. In one embodiment, media hardware output device 570 can be a S/PDIF (Sony/Phillips Digital Interface) card for providing multiple outputs, (e.g., an analog output 573 and a digital output 575). An alternative media hardware output device providing similar digital output can also be implemented as device 570 including, but not limited to, a USB (universal serial bus) output device and/or an externally accessible USB port located on system 210, a FireWire (IEEE1394) output device and/or an externally accessible FireWire port located on system 210, with wireline or wireless communication functionality.

In one embodiment, CCM 300 is communicatively coupled with playback application 501 via coupling 520, waveform driver shim 309 via coupling 522, custom

media device 310, via coupling 525, and media hardware output device 570 via coupling 523. CCM 300 is coupled to and controls selectable switch 311 in waveform driver shim 309 via coupling 522. CCM 300 is also coupled to and controls selectable switch 312 in custom audio device 310 via coupling 525. CCM 300 is further coupled to and controls selectable switch 571 in media hardware output device 570 via coupling 523. Depending upon the copyright restrictions and licensing agreements applicable to an incoming media file, (e.g., media 499), CCM 300 controls whether switches 311 and 312 are open (shown), thus preventing the incoming media 499 from reaching a recording application, or closed (not shown) so as to allow recording of the incoming media 499. Additionally, CCM 300 controls whether switch 571 is open (shown), thus preventing incoming media 499 from being output from digital output 575 of media hardware output device 570, or closed (not shown) to allow incoming media 499 to be output from media hardware output device 570.

By controlling media hardware output device 570, copyright compliance mechanism 300 can prevent unauthorized output of incoming media 499 to, e.g., a digital recording device that may be coupled with digital output 575 of media hardware output device 570. Accordingly, in one embodiment, CCM 300 is enabled to also detect digital recording devices that may be coupled to a digital output line, e.g., 575, of a media hardware output device, (e.g., 570). Examples of a digital recording device that can be coupled to media hardware output device 570 includes, but is not limited to, mini-disc recorders, MP3 recorders, personal digital recorders, digital recording devices coupled with multimedia systems, personal communication devices, set-top boxes, and/or nearly any digital device that can capture incoming media 499 being output from media hardware output device 570, (e.g., a sound card, video card, etc.).

Within Figure 5C, direct sound 504 is shown coupled with custom media device 310 via wave-in line 553, instead of being coupled with media device driver 505 (Figure 5A). In one embodiment, custom audio device 310 mandates explicit selection through system 210, meaning that custom audio device 310 needs to be selected as a default driver of system 210. By virtue of having the selection of custom media device 310 as the default driver of system 210, the data path necessary for direct sound 504 to capture the media content can be selectively closed.

For example, incoming media 499 originating from nearly any source with reference to Figure 5A is received by media playback application 501 of system 210. Playback application 501 communicates to CCM 300, via coupling 520, to determine whether incoming media 499 is protected by any copyright restrictions and/or licensing agreements. Playback application 501 communicates with CCM 300 to control switch 311, 312, and 571 accordingly. In the present example, recording of incoming media 499 would violate applicable restrictions and/or agreements and therefore switch 312 is in an open position, such that the output path to recording application 502, (e.g., wave-out line 548 and/or wave-out line 568), is effectively blocked, thereby preventing unauthorized recording of media 499.

Alternatively, if media device driver 505 is selected as the default driver, incoming media 499 continues from O/S multimedia subsystem 503, through custom audio device 310, wave driver shim 309, and into media device driver 505 where media 499 can be simultaneously output to media output device 570 via line 580, and output on wave-out line 540 to wave shim driver 309 on wave-out line 542. However, by virtue

of CCM 300 controlling switch 311, wave-out line 544 which eventually leads to recording application 502 is blocked, thus effectively preventing unauthorized recording of media 499.

5 It is noted that by virtue of CCM 300 controlling both switches 311 and 312 and therefore controlling wave-out line 548 and wave-out line 568, any incoming media files, (e.g., 499), can be prevented from being recording in an unauthorized manner in accordance with applicable copyright restrictions and/or licensing agreements related to the incoming media.

10 Still referring to Figure 5C, it is noted that although CCM 300 can prevent unauthorized recording of incoming media 499 by controlling switches 311 and 312, thus preventing incoming media 499 from reaching recording application 502, controlling switches 311 and 312 do nothing to prevent incoming media 499 from being
15 captured by a peripheral digital device, (e.g., a mini-disc recorder, etc.), coupled to digital output 575 of device 570. Thus, by also controlling digital output 575 of media hardware output device 570 via switch 571, CCM 300 can prevent unauthorized capturing of incoming media 499 from output 575, (e.g., on a sound card for audio files, a video card for video and/or graphical files), regardless of whether incoming media
20 499 is received in a secure and encrypted manner. However, when switch 571 is in a closed position, incoming media 499 may be played back in an unfettered manner. Additionally, at any time during playback of media 499, switch 312 of custom media device 310, switch 311 of media device driver 309, and/or switch 571 of media hardware output device 570 can be dynamically activated by CCM 300.

Figure 5D is an exemplary logic/bit path block diagram 500D of a client computer system, (e.g., 210), configured with a copyright compliance mechanism 300 for preventing unauthorized kernel based output and unauthorized recording of copyrighted media according to an embodiment of the present invention. Copyright compliance mechanism 300 is, in one embodiment, coupled with and operational on client system 210 in the manner described herein with reference to Figures 4, 5A, 5B, 5C, 6, and 7.

Diagram 500D of Figure 5D is similar to diagram 500C of Figure 5C, with some changes. Particularly, diagram 500D includes a kernel streaming mechanism 515, (e.g., DirectKS), that is coupled with media device driver 505. In one embodiment, DirectKS 515 can be used for establishing a direct connection with media device driver 505. In the present embodiment, media device driver 505 is shown to include a switch 511 controlled by CCM 300 via communication line 524, that is similar to switches 311, 312, and 571, for controlling output of incoming media 499.

In one embodiment, CCM 300 is communicatively coupled with: playback application 501 via coupling 520, waveform driver shim 309 via coupling 522, custom media device 310 via coupling 525, and media device driver 505 via coupling 524. Specifically, CCM 300 is coupled to and controls selectable switch 311 of waveform driver shim 309 via coupling 522. CCM 300 is also coupled to and controls selectable switch 312 of custom audio device 310 via coupling 525. CCM 300 is further coupled to and controls selectable switch 511 of media device driver 505 via coupling 524. Depending upon the copyright restrictions and/or licensing agreements applicable to an incoming media file, (e.g., media 499), CCM 300 controls whether switches 311 and

312 are open (shown), thus preventing the incoming media 499 from reaching a recording application, or closed (not shown) so as to allow recording of the incoming media 499. Additionally, CCM 300 controls whether switch 511 is open (shown), thus preventing incoming media 499 from capturing incoming media 499 and redirecting it to recording application 502 to create an unauthorized copy or recording of incoming media 499. CCM 300 can also control whether switch 511 is closed (not shown) to allow DirectKS 515 to capture and redirect incoming media 499 to recording application 502.

DirectKS 515, in one embodiment, may represent a kernel streaming mechanism that is adapted to establish a direct connection with media device driver 505 of an operating system operable on client computer system 210, enabling kernel level access to media device driver 505. A kernel streaming mechanism can be implemented for the purpose of precluding utilization of standard audio APIs (application programming interfaces) to play or record media content, with particular attention paid to those playback applications with low latency requirements. DirectKS 515 can bypass existing APIs and communicate with media device driver 505. DirectKS 515 can be readily adapted to work in conjunction with a playback application, (e.g., 501), via coupling 581 to capture incoming media 499 and redirect it to driver 505 via coupling 583 and then to recording application 502 via wave-out line 588. Accordingly, DirectKS 515 can be implemented to create unauthorized media recordings.

By controlling media device driver 505, copyright compliance mechanism 300 can prevent unauthorized output of incoming media 499 to, e.g., a digital recording device 529 that may be coupled with recording application 502. In one embodiment, media device driver 505 is configured through the kernel mixer (not shown) to control

the data path. Additionally, in one embodiment, CCM 300 is enabled to also detect a kernel streaming mechanism 515 (e.g., DirectKS) that may be operable on client computer system 210, as described herein with reference to Figure 3.

5 In one embodiment, custom media device 310 mandates explicit selection through system 210, meaning that custom media device 310 needs to be selected as a default driver of system 210. By virtue of having the selection of custom media device 310 as the default driver of system 210, the data path necessary for direct sound 504 to capture the media content is selectively closed.

10 For example, incoming media 499 originating from nearly any source described herein with reference to Figure 5A is received by media playback application 501 of system 210. Playback application 501 communicates to CCM 300, via connection 520, to determine whether incoming media 499 is protected by any copyright restrictions
15 and/or licensing agreements. Playback application 501 communicates with CCM 300 to control switches 311, 312, 571, and 511, accordingly. In the present example, recording of incoming media 499 would violate applicable restrictions and/or agreements and there (e.g., wave-out line 548 and/or wave-out line 568 and/or wave-out line 588), is effectively blocked, thereby preventing unauthorized recording of media 499.

20 Still referring to Figure 5D, it is particularly noted that although CCM 300 can prevent unauthorized recording of incoming media 499 by controlling switches 311, 312, and 571, thus preventing incoming media 499 from reaching recording application 502, controlling switches 311, 312, and 571, do nothing to prevent incoming media 499
25 from being returned to recording application 502 by a kernel streaming mechanism 515

(e.g., DirectKS), which enables capturing and redirecting of incoming media 499 to recording application 502, via wave-out line 588. Thus, by also controlling switch 511 of media device driver 505, CCM 300 can prevent kernel streaming mechanism 515 from returning incoming media 499 to recording application 502, thereby preventing incoming media 499 from being captured and redirected to recording application 502 in an attempt to create an unauthorized copy and/or recording of incoming media 499. However, when switch 511 is in a closed position, incoming media 499 may be returned to recording application 502, such that recording could be possible, provided recording does not violate copyright restrictions and/or licensing agreements applicable to incoming media 499. Additionally, at any time during playback of media 499, switch 312 of custom media device 310, switch 311 of wave shim driver 309, and/or switch 511 of media device driver 505 can be dynamically activated by CCM 300.

Figure 6 is a block diagram of a media file, (e.g., 499), adapted to be received by a playback application, (e.g., 501 of Figures 5A-5D), configured with an indicator 605 for enabling incoming media 499 to comply with rules according to the SCMS (serial copy management system). When applicable to a media file, e.g., 499, the SCMS allows for one copy of a copyrighted media file to be made, but not for copies of copies to be made. Thus, if incoming media 499 can be captured by a recording application, (e.g., 501 of Figures 5A-5D), and/or a recording device, (e.g. 529), and/or a peripheral recording device and/or a recording application coupled to a digital output of a media hardware output device, (e.g., digital output 575 of media hardware output device 570 of Figures 5B, 5C, and 5D), and/or a kernel streaming mechanism 515, (e.g., DirectKS 515 of Figure 5D), unauthorized copying and/or recording may be accomplished.

Playback application 501 is coupled with CCM 300 via communication line 520 in a manner analogous to Figures 5A, 5B, 5C, and/or 5D. Although not shown in Figure 6, it is noted that CCM 300 is also coupled to switches 311 and 511 as shown in Figure 5A, switches 311 and 312 in Figure 5B, switches 311, 312, and 571 in Figure 5C, and switches 312, 311, 571, and 511, in Figure 5D.

In one embodiment, an indicator 605 is attached to incoming media 499 for preventing unauthorized copying or recording in accordance with the SCMS. In one embodiment, indicator 605 can be a bit that may be transmitted prior to beginning the delivery of incoming media 499 to playback application 501. In another embodiment, indicator 605 may be placed at the beginning of the bit stream of incoming media 499. In yet another embodiment, indicator 605 may be placed within a frame period of incoming media 499, (e.g., every fifth frame), or any other desired frame period. In another embodiment, indicator 605 may be transmitted at a particular time interval or intervals during delivery of the media file, (e.g., 499). Thus, indicator 605 may be placed nearly anywhere within or attached to the bit stream related to incoming media 499.

Within Figure 6, indicator 605 may be comprised of various indicators, (e.g., a level 0 indicator, a level 1 indicator, and a level 2 indicator), in one embodiment of the present invention. In the present embodiment, a level 0 indicator may be for indicating to CCM 300 that copying is permitted without restriction, (e.g., incoming media 499 is not copyrighted or the copyright is not asserted). In the present embodiment, a level 1 indicator may be for indicating to CCM 300 that one generation of copies of incoming media 499 may be made, such that incoming media 499 is an original copy and that one

copy may be made. In the present embodiment, a level 2 indicator may be for indicating to CCM 300 that incoming media 499 is copyright protected and/or a copy thereof, and as such no digital copying is permitted.

5 For example, incoming media 499 is received by playback application 501. Application 501 detects an indicator 605 attached therewith, in this example, a level 2 bit placed in the bit stream indicates to CCM 300 that copying is not permitted. As such, when CCM 300 is configured in system 210 such as that shown in Figure 5A, in response to a level 2 indicator bit, CCM 300, while controlling the media path, then
10 activates switches 311 and 511 to prevent any recording of incoming media 499.

 However, CCM 300 is configured in system 210 such as that shown in Figure 5B, in response to a level 2 indicator bit, CCM 300, while controlling the media path, then activates switches 311 and 312 to prevent any recording of incoming media 499.
15

 Alternatively, when CCM 300 is configured in system 210 such as that shown in Figure 5C, in response to a level 2 indicator bit, CCM 300, while controlling the media path then activates switches 311, 312, and 571 to prevent any recording of incoming media 499.
20

 It is noted that CCM 300 can activate or deactivate switches coupled therewith, as described herein with reference to Figures 5A-5D, thereby funneling incoming media 499 through the secure media path, in this instance the audio path, to prevent unauthorized copying of incoming media 499. It is further noted that CCM 300 can

detect media recording applications and devices as described herein, with reference to Figure 3.

Figures 7A, 7B, and 7C, are a flowchart 700 of steps performed in accordance with one embodiment of the present invention for controlling end user interaction of delivered electronic media. Flowchart 700 includes processes of the present invention which, in some embodiments, are carried out by processors and electrical components under the control of computer readable and computer executable instructions. The computer readable and computer executable instructions reside, for example, in data storage features such as computer usable volatile memory 102 and/or computer usable non-volatile memory 103 of Figure 1. However, the computer readable and computer executable instructions may reside in any type of computer readable medium. Although specific steps are disclosed in flowchart 700, such steps are exemplary. That is, the present embodiment is well suited to performing various other steps or variations of the steps recited in Figures 7A, 7B, and 7C. Within the present embodiment, it should be appreciated that the steps of flowchart 700 may be performed by software, by hardware or by any combination of software and hardware.

The present embodiment provides a method for restricting recording of high fidelity media content delivered via one or more communication networks. The present embodiment delivers the high fidelity media content to registered clients while preventing unauthorized clients from directly receiving media content from a source database. Once the client computer system receives the media content, it can be stored in hidden directories and/or custom file systems that may be hidden to prevent subsequent unauthorized sharing with others. It is noted that various functionalities

can be implemented to protect and monitor the delivered media content. For example, the physical address of the media content can be hidden from media content recipients. Alternatively, the directory address of the media content can be periodically changed. Additionally, an access key procedure and rate control restrictor can also be
5 implemented to monitor and restrict suspicious media content requests. Furthermore, a copyright compliance mechanism, (e.g., CCM 300), can be installed in the client computer system 210 to provide client side compliance with licensing agreements and/or copyright restrictions applicable to the media content. By implementing these and other functionalities, the present embodiment restricts access to and the
10 distribution of delivered media content and provides a means for copyrighted media owner compensation.

It is noted that flowchart 700 is described in conjunction with Figures 2, 3, 4, and 5A-5D, in order to more fully describe the operation of the present embodiment. In
15 operation 702 of Figure 7A, a user of a computer system, (e.g., 210), causes the computer to communicatively couple to a web server, (e.g., 250), via one or more communication networks, (e.g., Internet 201), and proceeds to attempt to log in. It is understood that the log in process of step 702 can be accomplished in a variety of ways in accordance with the present invention.

20 In operation 704 of Figure 7A, web server 250 accesses a user database, (e.g., 450), to determine whether the user and the computer system 210 logging in are registered with it. If the user and computer system 210 are registered with web server 250, the present embodiment proceeds to operation 714. However, if the user and computer

system 210 are not registered with web server 250, web server 250 can initiate a user and computer system 210 registration process at operation 706.

In operation 706, registration of the user and computer system 210 is initiated.

5 The user and computer system registration process can involve the user of computer system 210 providing personal information including, but not limited to, their name, address, phone number, credit card number, online payment account number, biometric identification (e.g., fingerprint, retinal scan, etc.), and the like. Web server 250 can verify the accuracy of the information provided. Web server 250 can also acquire
10 information regarding the user's computer system 210 including, but not limited to, identification of media players disposed and operable on system 210, a unique identifier corresponding to the computer system, etc. In one embodiment, the unique identifier corresponding to the computer system can be a MAC address. Additionally, web server 250 can further request that the user of computer system 210 select a username
15 and password.

In operation 708 of Figure 7A, subsequent to the completion of the registration process, web server 250 generates a unique user identification (ID) or user key associated with the user of client computer system 210. The unique user ID, or user
20 key, is then stored by web server 250 in a manner that is associated with that registered user. Furthermore, one or more cookies containing that information specific to that user and the user's computer system 210, is installed in a non-volatile memory device, (e.g., 103 and/or data storage device 108 of computer system 210). It is noted that the user ID and cookie can be stored in a hidden directory within one or more non-volatile memory
25 devices within computer system 210, thereby preventing user access and/or

manipulation of that information. It is further noted that if the unique user ID, or user key, has been previously generated for the user and computer 210 that initially logged-in at operation 702, the present embodiment proceeds to operation 714

5 In operation 710, web server 250 verifies that the user ID and the cookie(s) are properly installed in computer system 210 and verifies the integrity of the cookie(s) and the user ID, thereby ensuring no unauthorized alterations to the user ID or the cookie(s) has occurred. If the user ID is not installed and/or not valid, web server 250 can re-initiate the registration process at operation 706. Alternatively, web server 250 can
10 decouple computer system 210 from the network, thereby requiring a re-log in by the user of computer 210. If the cookie(s) and user ID are valid, the present embodiment proceeds to operation 712.

 In operation 712 of Figure 7A, web server 250 can install a version of a copyright
15 compliance mechanism, (e.g., 300), onto one or more non-volatile memory devices of computer system 210. Installing CCM 300 into user's computer system 210 can facilitate client side compliance with licensing agreements and copyright restrictions applicable to specific delivered copyrighted media content. At operation 712, the components of CCM 300, such as instructions 301, coder/decoder (codec) 303, agent programs 304,
20 system hooks 305, skins 306, and custom media device drivers 307 (e.g., custom media device 310 of Figures 5B-5D), are installed in computer system 210, such as that shown in Figures 5A-5D. In one embodiment, a hypertext transfer protocol file delivery system can be utilized to install CCM 300 into computer system 210. However, operation 712 is well suited to install CCM 300 on computer system 210 in a wide
25 variety of ways in accordance with the present embodiment. For example, CCM 300

can be installed as an integrated component within a media player application, media recorder application, and/or media player/recorder application. Alternatively, CCM 300 can be installed as a stand-alone mechanism within client computer system 210.

Additionally, CCM 300 can be installed as a stand-alone mechanism and/or as part of a
5 bundled application from a media storage device, (e.g., a CD, a DVD, an SD), and/or as part of an installation package. In another embodiment, CCM 300 can be installed in conjunction with a presentation of desired media content, (e.g., listening to an audio file on a music CD, reading a document, viewing a video, etc.). It is noted that, in one embodiment, CCM 300 may be installed on client system 210 in a clandestine manner,
10 relative to a user.

In operation 714, web server 250 can request the previously established username and password of the user of client computer system 210. Accordingly, the user of client computer system 210 causes it to transmit to web server 250 the
15 previously established username and password. Upon the receipt thereof, web server 250 may access a user database, (e.g., 450), to determine their validity. If the username and password are invalid, web server 250 refuses access wherein flowchart 700 may be discontinued (not shown). Alternatively, if the username and password are valid, the present embodiment proceeds to operation 716.

20 In operation 716 of Figure 7A, web server 250 can access media file database 450 to determine if copyright compliance mechanism 300 has been updated to reflect changes made to the DMCA (Digital Millennium Copyright Act) and/or to the interactive/non-interactive licensing agreements recognized by the DMCA. It is noted
25 that alternative licensing agreements can be incorporated into copyright compliance

mechanism 300. Advantageously, by providing a copyright compliance mechanism that can be readily updated to reflect changes in copyright restrictions, licensing agreements, and/or changes to existing media player applications, and/or the development of new media player applications, copyright compliance mechanism 300
5 can provide compliance with current restrictions associated with the media content.

Continuing with operation 716, if web server 250 determines that CCM 300, or components thereof, of computer 210 has not been updated, web server 250 initiates installation of the newer components and/or the most current version of CCM 300 into
10 computer system 210, shown as operation 718. If web server 250 determines that the current version of CCM 300 installed on system 210 does not have to be updated, the present embodiment proceeds to operation 720 of Figure 7B.

In operation 720 of Figure 7B, the user of client computer system 210 causes it to
15 transmit to web server 250, (e.g., via Internet 201), a request for a play list of available media files. It is noted that the play list can contain all or part of the media content available from a content server, (e.g., 251).

In operation 722, in response to web server 250 receiving the play list request,
20 web server 250 transmits to client computer system 210 a media content play list together with the unique user ID associated with the logged-in user. The user ID, or user key, can be attached to the media content play list in a manner invisible to the user. It is noted that the media content in content server 251 can be, but is not limited to, high fidelity music, audio, video, graphics, multimedia, alphanumeric data, and the like.
25 The media content play list of operation 720 can be implemented in diverse ways. In

one example, web server 250 can generate a media content play list by combining all the available media content into a single play list. Alternatively, all of the media content titles, or different lists of titles, can be loaded from content server 251 and passed to a CGI (common gateway interface) program operating on web server 250 where the media titles, or differing lists of titles, can be concatenated into a single dimensioned array that can be provided to client computer system 210. It is understood that the CGI can be written in nearly any software computing language.

In operation 724 of Figure 7B, the user of client computer system 210 can utilize the received media content play list in conjunction with a media player application in order to cause client computer system 210 to transmit a request to web server 250 for delivery of desired media content, and wherein the user ID is automatically included therewith. The media content play list provided to client computer system 210 by web server 250 can enable the user to create one or more customized play lists by the user selecting desired media content titles. It is noted that a customized media play list can establish the media content that will eventually be delivered to client computer system 210 and the order in which the content will be delivered. Additionally, the user of client computer system 210 can create one or more customized play lists and store those play lists in system 210 and/or within web server 250. It is noted that a customized play list does not actually contain the desired media content titles, but rather the play list includes one or more identifiers associated with the desired media content that can include, but is not limited to, a song, an audio clip, a video clip, a picture, a multimedia clip, an alphanumeric document, or particular portions thereof. In another embodiment, the received media content play list can include a random media content

delivery choice that the user of client computer system 210 can transmit to web server 250, with the user ID, to request delivery of the media content in a random manner.

In operation 726, upon receiving the request for media content from client
5 computer system 210, web server 250 determines whether the requesting media application operating on client computer system 210 is a valid media application. One of the functions of a valid media application is to be a player of media content as opposed to an application that downloads media content in an unauthorized or unregulated manner. If web server 250 determines that the media application operating
10 on system 210 is not a valid media application, the present embodiment proceeds to operation 727 which in one embodiment, redirects client computer 210 to a web site where the user of system 210 can download a valid media player application or to a software application which can identify client computer system 210, log system 210 out of web server 250 and/or prevent future logging-in for a defined period of time, (e.g.,
15 15 minutes, an hour, a day, a week, a month, a year, or any specified amount of time). If web server 250 determines that the media application operating on system 210 is a valid media application, the present embodiment proceeds to operation 728.

In operation 728 of Figure 7B, the present embodiment causes web server 250 to
20 determine whether the user ID (or user key) that accompanied the media delivery request sent by client computer system 210 is valid. If web server 250 determines that the user ID is invalid, the present embodiment proceeds to operation 729 where client computer system 210 can be logged off web server 250 or client computer system 210 can be returned to operation 706 (of Figure 7A) to re-register and to have another
25 unique user ID generated by web server 250. It is noted that the order in which

operation 726 and 728 are performed can be altered such that operation 728 can be performed prior to operation 726. If web server 250 determines that the user ID is valid, the present embodiment proceeds to operation 730.

5 In operation 730, prior to web server 250 authorizing the delivery of the redirect and access key for the requested media file content, shown as operation 732, CCM 300 governs certain media player applications and/or functions thereof that are operable on client computer system 210. These governed functions can include, but is not limited to, pause, stop, progress bar, save, etc. It is noted that, in one embodiment, CCM 300 can
10 utilize system hooks 305 to accomplish the functionality of operation 730.

 In operation 732 of Figure 7C, the present embodiment causes web server 250 to transmit to client computer system 210 a redirection command along with a time sensitive access key (e.g., for that hour, day or for any defined period of time) thereby
15 enabling client computer 210 to receive the requested media content. The redirection command can include a time sensitive address of the media content location within content server 251. The address is time sensitive because, in one embodiment, the content server 251 periodically renames some or all of the media address directories, thereby making previous content source addresses obsolete. Alternatively, the address
20 of the media content is changed. In another embodiment, the location of the media content can be changed along with the addresses. Regardless, unauthorized users and/or applications are restricted from directly retrieving and/or copying the media content from content server 251. Therefore, if someone with inappropriate or unlawful intentions is able to find where the media content is stored, subsequent attempts will

fail, as the previous route no longer exists, thereby preventing future unauthorized access.

It is noted that in one embodiment of the present invention, the addresses (or
5 routes) of content server 251 that are actively coupled to one or more client computer
systems (e.g., 210 -230) are maintained while future addresses, or routes, are being
created for new client devices. It is further noted that as client computer systems are
uncoupled from the media content source of content server 251, that directory address,
or link, can be immediately changed, thereby preventing unauthorized client system or
10 application access.

In another embodiment, the redirection of client computer system 210 to content
server 251 can be implemented by utilizing a server network where multiple servers are
content providers, (e.g., 251), or by routing a requesting client computer system (e.g.,
15 210, 220, or 230) through multiple servers. In yet another embodiment, the delivery of
media content from a central content provider (e.g., 251) can be routed through one or
more intermediate servers before being received by the requesting client computer
system, (e.g., 210).

20 The functionality of operation 732 is additionally well suited to provide
recording of the Internet Protocol (IP) addresses of the client computer systems, (e.g.,
210), the media content requested and its transfer size, thereby enabling accurate
monitoring of royalty payments, clock usage and transfers, and media content
popularity.

25

In operation 734 of Figure 7C, upon receiving the redirection command, the present embodiment causes the media playback application 501 (Figures 5A-5D) operating on client computer system 210 to automatically transmit to content server 251 a new media delivery request which can include the time sensitive access key and the address of the desired media content.

In operation 736 of Figure 7C, content server 251 determines whether the time sensitive access key associated with the new media delivery request is valid. If content server 251 determines that the time sensitive access key is valid, the present embodiment proceeds to operation 738 of Figure 7C. However, if content server 251 determines that the time access key is not valid, the present embodiment proceeds to operation 737, a client redirect.

In operation 737, content server redirects client computer 210 to operation 732 where a new access key is generated. Alternatively, operation 737 causes the present embodiment to return to operation 704 of Figure 7A. In yet another embodiment, operation 737 can cause client computer system 210 to be disconnected from content server 251.

In operation 738 of Figure 7C, content server 251 transmits the requested high fidelity media content to client computer system 210. It is noted that each media content file delivered to client computer system 210 can have a header attached thereto, prior to delivery, as described herein with reference to Figure 4. It is further noted that both the media content and the header attached thereto can be encrypted. In one embodiment, the media content and the header can be encrypted differently.

Alternatively, each media content file can be encrypted differently. In another embodiment, groups of media files are analogously encrypted. It is noted that public domain encryption mechanisms, (e.g., Blowfish), and/or non-public domain encryption mechanisms can be utilized.

5

Still referring to operation 738, content server 251 can transmit the requested media content in a burst load (in comparison to a fixed data rate), thereby transferring the content to client computer system 210 as fast as the network transfer rate allows. Further, content server 251 can have its download rate adapted to be equal to the transfer rate of the network to which it is coupled. In another embodiment, the content server 251 download rate can be adapted to equal the network transfer rate of the client computer system 210 to which the media content is being delivered. For example, if client computer system 210 is coupled to Internet 201 via a T1 connection, then content server 251 transfers the media content at transmission speeds allowed by the T1 connection line. As such, once the requested media content is transmitted to client computer system 210, content server 251 is then able to transmit requested media content to another client computer system, (e.g., 220 or 230). Advantageously, this provides an efficient means to transmit media content, in terms of statistical distribution over time and does not overload the communication network(s).

20

It is noted that delivery of the requested media content by content server 251 to client computer system 210 can be implemented in a variety of ways. For example, an HTTP (hypertext transfer protocol) file transfer protocol can be utilized to transfer the requested media content as well as copyright compliance mechanism 300 to client 210.

25 In this manner, the copyright compliance mechanism as well as each media content

file/title can be delivered in its entirety. In another embodiment, content server 251 can transmit to client computer system 250 a large buffer of media content, (e.g., audio clips, video clips, and the like).

5 In operation 740 of Figure 7C, upon receiving the requested high fidelity media content from content server 251, the present embodiment causes client computer system 210 to store the delivered media content in a manner that is ready for presentation, (e.g., playback). The media content is stored in client computer system 210 in a manner that restricts unauthorized redistribution. For example, the present embodiment can cause
10 the high fidelity media content to be stored in a volatile memory device (e.g., 102), utilizing one or more hidden directories and/or custom file systems that may be hidden, where it may be cached for a limited period of time. Alternatively, the present embodiment can cause the high fidelity media content to be stored in a non-volatile memory device, (e.g., 103) or data storage device (e.g., 108). It is noted that the manner
15 in which each of the delivered media content file(s) is stored, volatile or non-volatile, can be dependent upon the licensing restrictions and/or copyright agreements applicable to each media content file. It is further noted that in one embodiment, when a user of client computer system 210 turns the computer off or causes client computer system 210 to disconnect from the network, the media content stored in a volatile
20 memory device is typically deleted therefrom.

 Still referring to operation 740, in another embodiment, the present embodiment can cause client computer system 210 to store the received media content in a non-volatile manner within a media application operating therein, or within one of its
25 Internet browser applications (e.g., Netscape Communicator™, Microsoft Internet

Explorer™, Opera™, Mozilla™, and the like) so that delivered media content can be used in a repetitive manner. Further, the received media content can be stored in a manner making it difficult for a user to redistribute in an unauthorized manner, while allowing the user utilization of the received media content, (e.g., by utilizing one or more hidden directories and/or custom file systems that may also be hidden). It is noted that by storing media content with client computer system 210 (when allowed by applicable licensing agreements and/or copyright restrictions), content server 251 does not need to redeliver the same media content to client computer 210 each time its user desires to experience (e.g., listen to, watch, view, etc.) the media content file.

In operation 742 of Figure 7C, the received media content file is then fed into a media player application (e.g., 501 of Figures 5A-5D), which then runs it through a codec, (e.g., 303 of CCM 300), in one embodiment. In response, coder/decoder 303 sends an authorization request to the content server, (e.g., 251), with attached authorization data, as described herein. In response to receiving codec's 303 authorization request, server 251 compares the received authorization data with that stored in server 251, and subsequently, the present embodiment proceeds to operation 744.

In operation 744, the content server 251 responds with a pass or fail authorization. If server 251 responds with a fail, such that the received authorization data is invalid, the present embodiment can proceed to operation 745, where server 251 can, in one embodiment, notify the user of client system 210, (e.g., by utilization of skin 306), that there was an unsuccessful authorization of the requested media content file.

It is noted that alternative messages having similar meanings may also be presented to

the user of client computer system 210, thereby informing the user that the delivery failed. However, if the authorization data passes, the present embodiment proceeds to operation 746.

5 In operation 746, content server 251 transmits certain data back to the media player application enabling the media player application to present the contents of the media file via media playback application 501 of Figures 5A-5D. In one embodiment, a decryption key can be included in the transmitted data to decrypt the delivered media content file. In another embodiment, an encryption/decryption key can be included in
10 the transmitted data to allow access to the contents of the media file.

 In operation 748 of Figure 7C, subsequent to media file decryption, the media file may be passed through CCM 300, (e.g., a codec 303), to a media player application operating on client computer system 210, (e.g., playback application 501 of Figures 5A-
15 5D), which can then access and utilize the delivered high fidelity media content, enabling its user(s) to experience the media content, (e.g., listen to it, watch it, view it, or the like). In one embodiment of the present invention, a specialized or custom media player may be involved in order to experience the media content, (e.g., skin 306 of Figure 3). Skin 306 may be implemented when CCM 300 cannot modify an industry
20 standard media player application to comply with copyright restrictions and/or licensing agreements in accordance with the DMCA. Alternatively, a specialized or custom media player may not be needed to experience the media content. Instead, an industry standard media player can be utilized by client computer system 210 to experience the media content. Typically, many media player applications are available
25 and can include, but are not limited to, Windows™ Media Player™ for PCs (personal

computers), iTunes™ Player or QuickTime™ for Apple computers, and XMMS player for computers utilizing a Linux operating system. Regardless of the media player application utilized, while the media file is passed to the media player application, e.g., in a frame by frame basis or in a buffer by buffer basis, coder/decoder 303 will
5 repeatedly ensure that CCM 300 rules are being enforced at any particular moment during media playback, shown as operation 750.

In operation 750, as the media file content is delivered to the media player application, (e.g., 501 of Figures 5A-5D), periodically, (e.g., after a specified number of
10 frames, after a defined period of time, or any desired time or data period), coder/decoder 303 repeatedly determines whether or not all the rules are enforced, in accordance with rules as defined by CCM 300. If the rules are not enforced, (e.g., change due to a user opening up a recording application (e.g., Total Recorder or alternative application)) the present method proceeds to operation 751. If the rules, in
15 accordance with CCM 300, are enforced, the present embodiment then proceeds to operation 752.

In operation 751 of Figure 7C, if the rules according to CCM 300 are not enforced, the presentation of the media content is, in one embodiment, suspended or halted. In
20 one embodiment, CCM 300 of Figure 5A can selectively control switches 311 and 511 to prevent output of incoming media 499 to a recording application 502 via wave shim driver 309 and direct sound 504 respectively, thus preventing unauthorized recording of incoming media 499. In another embodiment, CCM 300 of Figure 5B can selectively control switches 311 and 312 to prevent output of incoming media 499 to recording
25 application 502 via wave shim driver 309 and custom media device 310, thus preventing

unauthorized recording of incoming media 499. In yet another embodiment, CCM 300 of Figure 5C can selectively control switches 311, 312, to not only prevent incoming media 499 from being recorded in an unauthorized manner but can also selectively control switch 571 to prevent unauthorized output of incoming media 499 via digital output 575 of media hardware output device 570. In yet another embodiment, CCM 300 of Figure 5D can selectively control switches 311, 312, 571, and 511 to prevent kernel streaming mechanism 515, (e.g., DirectKS) which can establish a connection with media device driver 505 of Figure 5D, from capturing incoming media content and returning it to recording application to create an unauthorized recording of the media content. In one embodiment, incoming media 499 may not be output from digital output 575. In another embodiment, incoming media 499 may be output via digital output 575 but in an inaudible manner, (e.g., silence). In yet another embodiment, incoming media 499 can be audible but recording functionality can be disabled, such that the media content cannot be recorded.

In operation 752, if the rules are enforced in accordance with CCM 300, codec 303 retrieves a subsequent portion of the media content that is stored locally in client computer system 210. The present embodiment proceeds to operation 748 where the newly retrieved portion of the media file is then presented by the client's media player application. In this manner, the playback of the media content is constantly monitored by the present embodiment. Advantageously, by constant monitoring playback media files, CCM 300 can detect undesired activities and enforce those rules defined by CCM 300. It is noted that process 700 can be exited (not shown) once the media file contents are presented in their entirety.

Figure 8 is a diagram of an exemplary high-speed global media content delivery system 800, in accordance with an embodiment of the present invention. In one embodiment, system 800 can be utilized to globally deliver media content, (e.g., audio media, video media, graphic media, multimedia, alphanumeric media, etc.), to one or more client computer systems, e.g., 210, 220, and/or 230, in conjunction with a manner of delivery similar to that described herein. In one embodiment, system 800 includes a global delivery network 802 that can include multiple content servers, (e.g., 804, 806, 808, 810, 812, 814, and 816), that can be located throughout the world and which may be referred to as points of presence or media delivery point(s). Each content server 804-816 can store a portion, a substantial portion, or the entire contents of a media content library that can be delivered to client computer systems via one or more networks, (e.g., LAN Internet 201, or a wide area network (WAN)). Accordingly, each content server 804-816 can provide media content to client computer systems in its respective vicinity of the world. Alternatively, each content server can provide media content to a substantial number of client computer systems.

For example, a media delivery point (MDP) 816, located in Tokyo, Japan, is able to provide and deliver media content from the media content library stored in its content database, (e.g., 451), to client computer systems within the Asiatic regions of the world while a media delivery point 812, located in New York City, New York, USA, is able to provide and deliver media content from its stored media content library to client devices within the Eastern United States and Canada. It is noted that each city name, (e.g., London, Tokyo, Hamburg, San Jose, Dallas, Amsterdam, or New York City), associated with one of the media delivery points 804-816 represents the location of that particular media delivery point or point of presence. However, it is further noted that

these city names are exemplary because media delivery points 804-816 can be located anywhere within the world, and as such are not limited to the cities shown in global network 802.

5 Still referring to Figure 8, it is further noted that global system 802 is described in conjunction with Figures 2, 3, 4, 5A-D, and 6, in order to more fully describe the operation of the present embodiment. Particularly, subsequent to a client computer system, (e.g., 210 of Figure 2), interacting with a web server, (e.g., 250 of Figure 2), as described herein, web server 250, in one embodiment, can redirect client computer
10 system 210 to receive the desired media content from an MDP (e.g., 804-816) based on one or more differing criteria.

For example, computer system 210 may be located in Brattleboro, Vermont, and its user causes it to log-in with a web server 250 which can be located anywhere in the
15 world. It is noted that operations 702-730 of Figures 7A and 7B can then be performed as described herein such that the present embodiment proceeds to operation 732 of Figure 7C. At operation 732, the present embodiment can determine which media delivery points, (e.g., 804, 806, 808, 810, 812, 814, or 816), can subsequently provide and deliver the desired media content to client computer system 210.

20

Still referring to Figure 8, one or more differing criteria can be utilized to determine which media delivery point (e.g., 804-816) to select for delivery of the desired media content. For example, the present embodiment can base its determination upon which media delivery point is in nearest proximity to client computer system 210, (e.g.,
25 media delivery point 812). This can be performed by utilizing the stored registration

information, (e.g., address), provided by the user of client computer system 210.

Alternatively, the present embodiment can base its determination upon which media delivery point provides media content to the part of the world in which client computer system is located. However, if each media of the delivery points (e.g., 804-816) stores differing media content, the present embodiment can determine which one can actually provide the desired media content. It is noted that these are exemplary determination criteria and the embodiments of the present invention are not limited to such implementation.

Subsequent to determination of which media delivery point is to provide the media content to client computer system 210 at operation 732, web server 250 transmits to client computer system 210 a redirection command to a media delivery point/content server, (e.g., 812), along with a time sensitive access key, also referred to as a session key, (e.g., for that hour, day, or any defined time frame) thereby enabling client computer system 210 to eventually receive the requested media content. Within system 800, the redirection command can include a time sensitive address of the media content location within media delivery point 812. Accordingly, the New York City media delivery point 812 can subsequently provide and deliver the desired media content to client computer system 210. It is noted that operation 732-742 of Figure 7C can be performed by media delivery point 812 in a manner similar to content server 251 described herein.

Advantageously, by utilizing multiple content servers, (e.g., media delivery point 804-816), to provide high fidelity media content to client computer systems, (e.g., 210-230), located throughout the world, communication network systems of the Internet

201 do not become overly congested. Additionally, global network 802 can deliver media content to a larger number of client computer systems (e.g., 210-230) in a more efficient manner. Furthermore, by utilizing communication technology having data transfer rates of up to 320 Kbps (kilobits per second) or higher, embodiments of the present invention provide for rapid delivery of the media content in a worldwide implementation.

Referring still to Figure 8, it is noted that media delivery points/content servers 804-816 of global network 802 can be coupled in a wide variety of ways in accordance with the present embodiment. For example, media delivery point 804-816 can be coupled utilizing wired and/or wireless communication technologies. Further, it is noted that media delivery points 804-816 can be functionally coupled such that if one of them fails, another media delivery point can take over and fulfill its functionality. Additionally, one or more web servers similar to web server 250 can be coupled to global network 802 utilizing wired and/or wireless communication technologies.

Within system 800, content server/media delivery point 804 includes a web infrastructure that, in one embodiment, is a fully redundant system architecture. It is noted that each of the MDP/content servers 806-816 of global network 802 can be implemented to include a web infrastructure in a manner similar to the implementation shown in MDP 804.

Specifically, the web infrastructure of media delivery point 804 includes firewalls 818 and 820 which are each coupled to global network 802. Firewalls 818 and 820 can be coupled to global network 802 in diverse ways, (e.g., utilizing wired and/or wireless

communication technologies). Particularly, firewalls 818 and 820 can each be coupled to global network 702 via a 10/100 Ethernet handoff. However, system 800 is not limited in any fashion to this specific implementation. It is noted that firewalls 818 and 820 are implemented to prevent malicious users from accessing any part of the web infrastructure of media delivery point 804 in an unauthorized manner. Additionally, firewall 818 can include a device 836, (e.g., a router or other switching mechanism), coupled therewith and a DB (database) server 840 coupled to device 836 while firewall 820 includes a device 838, (e.g., a router or other switching mechanism), coupled therewith and a DB (database) server 842 coupled to device 838. Furthermore, DB server 840 is coupled with device 838 and DB server 842 is coupled with device 836.

Still referring to Figure 8, and within media delivery point 804, firewall 818 is coupled to a director device 822 which is coupled to internal web application server 826 and 828, and a hub server 830. Firewall 820 is coupled to a director 824 which is coupled to internal web application servers 826 and 828, and hub server 830. Hub server 830 can be implemented in a variety of ways including, but not limited to, as a Linux hub server. Hub server 830 is coupled to a data storage device 832 capable of storing media content. Data storage device 832 can be implemented in a variety of ways, e.g., as a RAID (redundant array of inexpensive/independent disks) appliance.

It is noted that media delivery points 804-816 can be implemented in any manner similar to content server 250 described herein. Additionally, media delivery points 804-816 of the present embodiment can each be implemented as one or more physical computing devices, (e.g., computer system 100 of Figure 1).

In another embodiment, CCM 300 can be adapted to be disposed on a media storage device, (e.g., 999 of Figures 10 and 11). Media storage device 999 can be, but is not limited to, a CD, a DVD, or other optical or magnetic storage device. By virtue of disposing a version of CCM 300 on a media storage device 999, embodiments of the present invention can provide copy protection for audio, video, multimedia, graphics, information, data, software programs, and other forms of media that may contain copyrighted material and which may be disposed on a media storage device. Alternatively, CCM 300 can be adapted to be installed on a computer system, (e.g., 210), via a media storage device 999 upon which it may be disposed.

Figure 9 is a block diagram of a copyright compliance mechanism/media storage device (CCM/MSD) 900, a version of CCM 300 adapted to be disposed on a media storage device, (e.g., 999 of Figures 10 and 11) in accordance with an embodiment of the present invention. It is noted that CCM 300 in CCM/MSD 900 is analogous to CCM 300 as described in Figures 3, 4, 5A-D, 6A and 7A-C. Further, CCM/MSD 900 can be readily updated in accordance with global delivery system 800, as described in Figures 7A-C.

In one embodiment, CCM/MSD 900 is adapted to provide stand-alone compliance with copyright restrictions and/or licensing agreements applicable to media files that may be disposed on a media storage device, (e.g., 999). In another embodiment, CCM/MSD 900 is adapted to be installed on a computer system, (e.g., 210) to provide compliance with copyright restrictions and/or licensing agreements applicable to media files as described in Figures 3, 4, 5A-D, 6A and 7A-C.

Referring to Figure 9, CCM/MSD 900 includes an autorun protocol component 910 for invoking automatic installation of CCM 300. To deter users from attempts at defeating various features inherent to CCM 300, (e.g., the autorun feature), CCM 300's monitoring program, agent program 304, verifies that those features that are to be operational are operational, and if not, CCM 300 prohibits the user from experiencing the contents of the media storage device.

If a user somehow defeats the autorun feature, and the user attempts to utilize an application to capture an image of the content, the application will make an image of the content on the media storage device, which also images the copyright protection contained thereon. As such when the image is played, CCM 300 recognizes the copy protection is present, and CCM 300 will only allow the user to experience the content when authorized, once CCM 300 is installed.

By virtue of the protections as described above provided by CCM 300, users will be able to experience the content of the media storage device in the content's original high quality format, thereby obviating the need to compress the media file used on client system 210.

Advantageously, the user will no longer need to suffer through poor quality output as a result of severely compressed media files.

It is noted that when adapted to be implemented in conjunction with a secure file format, meaning that the format of the file is, without proper authorization, non-morphogenic, embodiments of the present invention also provide effective compliance with copyright restrictions and/or licensing agreements with secure files formats. CCM

300 can control the types of file formats into which the media file can be transformed, (e.g., .wav, .mp3, etc.).

In one embodiment, the autorun feature associated with a media storage device
5 drive, (e.g., 1112 of Figure 10) of client system 210 is activated and operational. Alternatively, a notice of required autorun activation within client system 210 may be displayed on the media storage device and/or the case in which the media storage device is stored.

10 In another embodiment, if CCM 300 is present or if the user is coupled to a server, then messages containing instructions on how to activate the autorun feature of client system 210 may be presented to the user.

In one embodiment autorun protocol component 910 can detect media storage
15 device drives resident on a computer system, (e.g., 210).

The following C++ source code is an exemplary implementation of a portion of
autorun protocol component 910 for detecting media storage device drives residing and
operable on client computer system 210, according to one embodiment of the present
20 invention.

```
    if ( (dwRetVal = GetLogicalDrives())  
        != (DWORD) 0)  
    {  
25      /* initialize variables */  
      dwMask = (DWORD) 1;  
  
      /* initialize path to root of current drive */  
      _tcscpy(szDrive, _T("A:\\\\"));
```

```

    for (nIndex = 0, dwMask = (DWORD) 1;
        dwMask != (DWORD) 0;
        nIndex++, dwMask <<= 1)
5      {
        if ((dwRetVal & dwMask) != 0)
        {
            /* construct path to root of drive */
            szDrive[0] = (TCHAR) 'A' + nIndex;
10          if (GetDriveType(szDrive) == DRIVE_CDROM)
            {
                MessageBox((HWND) 0,
                           _T("CD-ROM drive found."),
15                szDrive,
                MB_OK);
            }
            else
            {
20                /* clear bit at current position */
                dwRetVal &= (~dwMask);
            }
        }
    }
25 }

```

In another embodiment, autorun protocol component 910 can detect whether a media storage device containing media files has been inserted into a media storage device drive coupled with client computer system 210, (e.g., drive 1112 of Figure 10). In another embodiment, CCM 300 can include instructions for monitoring media storage device drive 1112, and upon detection of drive activation, CCM 300 determines what type of media storage device has been inserted therein. Subsequently, CCM 300 can detect various triggers on the media storage device to invoke its protection, (e.g., a hidden file on newer media storage devices and/or the copyright indicator bit on legacy media storage devices), obviating the need for autorun. Upon detection, CCM 300 can invoke the appropriate protection for the associated media file.

The following C++ source code is an exemplary implementation of a portion of autorun protocol component 910 for detecting a media storage device inserted in a media storage device drive residing and operable on client computer system 210, according to one embodiment of the present invention.

```

5
    /* set error mode for operation */
    uiErrMode = SetErrorMode(SEM_FAILCRITICALERRORS);

    /* initialize path to root of current drive */
10    _tcscpy(szDrive, _T("A:\\"));

    for (nIndex = 0, dwMask = (DWORD) 1;
        dwMask != (DWORD) 0;
        nIndex++, dwMask <<= 1)
15    {
        if ((dwCDROMMask & dwMask) != 0)
        {
            /* construct path to root of drive */
            szDrive[0] = (TCHAR) 'A' + nIndex;
20
            if ( GetDiskFreeSpace(szDrive,
                                &dwSectors,
                                &dwBytes,
                                &dwClustersFree,
                                &dwClusters)
25
                != 0)
            {
                /* add bit for drive to mask */
                dwRetVal |= dwMask;
30            }
        }
    }

    /* restore original error mode */
35    SetErrorMode(uiErrMode);

```

Additionally, autorun protocol component 910 can also detect changes in media, (e.g., insertion of a different media storage device 999). Further, other media changes
40 can be detected subsequent to adaptation of the source code including, but not limited

to, detecting a previously accessed media file and/or detecting a previously inserted media storage device.

The following C++ source code is an exemplary implementation of a portion of autorun protocol component 910 for detecting a change in media, according to one embodiment of the present invention.

```

    /* initialize path to root of current drive */
    _tcscopy(szDrive, _T("A:\\"));
10  for (nIndex = 0, dwMask = (DWORD) 1;
        dwMask != (DWORD) 0;
        nIndex++, dwMask <= 1)
    {
15      /* check for presence of CD-ROM media in drive */
      if ((dwCurrMask & dwMask) != 0)
      {
          /* check if media previously in drive */
          if ((dwPrevMask & dwMask) == 0)
20          {
              /* construct path to root of drive */
              szDrive[0] = (TCHAR) 'A' + nIndex;

              /* check for presence of marker on drive */
              if (IsMPBMarkerPresent(szDrive) != 0)
25              {
                  /* process autorun information present on drive */
                  nRetVal = ProcessAutorun(szDrive);
              }
          }
30      }
    }
}
```

Still referring to Figure 9, CCM/MSD 900 also includes a kernel level filter driver 920 for controlling a data input path of an operating system coupled with and operable on client computer system 210.

CCM/MSD 900 also includes a generalized filter driver 930 for controlling ripping and "burning" applications, (e.g., Nero, Roxio, Exact Audio Copy, and others), thereby preventing such activities.

5

The following C++ source code is an exemplary implementation of a portion of generalized filter driver 930 for controlling ripping and burning applications that may be residing on and operable within client computer system 210, in accordance with one embodiment of the present invention.

10

```

bool  bDisabled;          /* flag indicating CD reads disabled */

    /* initialize variables */
    bDisabled = false;

15    if (bProtected == true)
    {
        if (type == IRP_MJ_DEVICE_CONTROL)
        {
20            ULONG ulIoControlCode = stack-
>Parameters.DeviceIoControl.IoControlCode;

            if (ulIoControlCode == IOCTL SCSI_PASS_THROUGH)
            {
25                SCSI_PASS_THROUGH * pspt = (SCSI_PASS_THROUGH *)
Irp->AssociatedIrp.SystemBuffer;

                if ( (pspt != NULL)
                    && (pspt->Cdb[0] == SCSIOP_READ_CD))
30                {
                    pspt->DataTransferLength = 0;
                    pspt->ScsiStatus = 0;

                    bDisabled = true;

35                }
            }
        }
        else if (ulIoControlCode ==
IOCTL SCSI_PASS_THROUGH_DIRECT)
        {

```

```

        SCSI_PASS_THROUGH_DIRECT * psptd =
(SCSI_PASS_THROUGH_DIRECT *)
Irp->AssociatedIrp.SystemBuffer;

5         if ( (psptd != NULL)
            && (psptd->Cdb[0] == SCSIOP_READ_CD))
        {
            psptd->DataTransferLength = 0;
            psptd->ScsiStatus = 0;
10         bDisabled = true;
        }
    }
15 }

    if (bDisabled == true)
    {
        /* complete current request */
20     status = CompleteRequest(Irp, STATUS_SUCCESS, 0);
    }
    else
    {
        /* pass request down without additional processing */
25     status = IoAcquireRemoveLock(&pdx->RemoveLock, Irp);

        if (!NT_SUCCESS(status))
            return CompleteRequest(Irp, status, 0);

30     IoSkipCurrentIrpStackLocation(Irp);
        status = IoCallDriver(pdx->LowerDeviceObject, Irp);
        IoReleaseRemoveLock(&pdx->RemoveLock, Irp);
    }
35

```

Still referring to Figure 9, CCM/MSD 900 includes a CCM 300, analogous to CCM 300 of Figure 3, that is adapted to be installed in client computer system 210 in one or more ways described herein.

40

In one embodiment, kernel level filter driver 920, generalized filter driver 930 and CCM 300 of CCM/MSD 900 are automatically installed on client computer system

210, subsequent to insertion of media storage device 999 into a media storage device drive, (e.g., 1112 of Figures 10 and 11. Autorun protocol component 910, as described above, detects insertion of media storage device 999 into an appropriate drive, and initiates installation of the components, (e.g., CCM 300, driver 920 and driver 930). In one embodiment, drivers 920 and 930 may be temporarily installed and may be deleted upon removal of media storage device 999 from media storage device drive 1112. In yet another embodiment, drivers 920 and 930 may be installed in hidden directories and/or files within client computer system 210. In another embodiment, some components of CCM 300 can remain installed on client system 210, (e.g. the monitoring program (agent program 304). In still another embodiment, other components, (e.g., the kernel level filter driver 920), can be dynamically loaded and unloaded as necessary in accordance with copyright restrictions and/or licensing agreements applicable to the media file.

Embodiments of the present invention utilize software, (e.g., CCM/MSD 900), that is placed on media storage device 999, in conjunction with controlling software CCM 300 installed on client computer system 210, and web server 250 and/or content server 251, wherein each component is communicatively coupled with the other via the Internet, thereby enabling dynamic updating of CCM 300 in the manner as described with reference to Figure 4, and operation 716 and 718 of Figures 7A-C.

In the present embodiment, CCM/MSD 900 provides a stand alone DRM that is far more sophisticated than existing DRM solutions. This is because CCM/MSD 900 goes into the data pathway of the operating system operable on client computer system 210 and obtains control of the data pathway, (e.g., filter driver 1108 of Figure 11), rather than exploiting inefficiencies or errors in the computer system.

Figure 10 is a block diagram of a communicative environment 1000 for controlling unauthorized reproduction of protected media files disposed on a media storage device in accordance with an embodiment of the present invention. Included in communicative environment 1000 is a media storage device drive 1112 coupled with a client computer system 210 via a data/address bus 110. Client computer system 210 is coupled with web server 250 and content server 251 via Internet 201. A media storage device 999, upon which a CCM/MSD 900 may be disposed, can be inserted in media storage device drive 1112. As such, autorun protocol component 910 detects the insertion and automatically invokes installation of CCM 300, kernel level filter driver 920 and generalized filter driver 930 from media storage device 999 into client computer system 210. Subsequent to installation, CCM 300 initiates a dynamic update with web server 250 and/or content server 251, via Internet 201. By installing CCM 300 on client computer system, agent program 304 (Figure 3) of CCM 300 is able to control the integrity of the software associated with CCM/MSD 900. Additionally, by conferring with servers 250 and/or 251 via Internet 201 online, the CCM 300 software version on media storage device 999 and installed on client computer system 210 can be updated when circumventions occur and/or kept current from platform to platform.

Advantageously, the monitoring mechanism of agent program 304 enables constant morphing of the version of CCM 300 disposed on media storage device 999 by communicating with server 250 and/or 251 and utilizing the dynamic update capabilities of global network 800 to readily update that which has been installed on client computer system 210, via media storage device 999.

In one embodiment, the installation is performed clandestine with respect to the user and is initiated by inserting media storage device 999 into an appropriate media storage device drive, (e.g. a magnetic/optical disk drive or alternative device drive coupled with client system 210). If the user is not registered with CCM 300, as
5 described herein with reference to Figure 4 and Figures 7A-7C, once installed, CCM 300 initiates an update process with web server 250 and/or content server 251 to readily include updates that have been invoked subsequent to release of the media file on media storage device 999. By virtue of the dynamic update capabilities of CCM 300, regardless of the version of CCM 300 on media storage device 999, CCM 300 provides
10 compliance with copyright restrictions and/or licensing agreements applicable to the media file on media storage device 999. Advantageously, enabling dynamic adaptability of CCM 300 provides for continued interoperability with new and updated operating systems, advancements in electronic technology, communication technologies and protocols, and the like, ensuring the effectiveness of CCM 300 into the future.

15 In another embodiment, if the user is a registered user with global delivery system 800, CCM 300 can detect which version is most current. Accordingly, when the version existing on client system 210 is more current than the version (for install) on media storage device 999, CCM 300 can bypass the install process and present the
20 contents contained on media storage device 999 to the user for them to experience.

Further advantageous, this technology is backward compatible with media storage device drives manufactured subsequent to and including the year 1982. Additionally, CCM 300 is compatible with media storage devices having a copyright

indicator bit disposed thereon. The copyright indicator bit has been included on all CDs released since the year 1982.

5 In the present embodiment of Figure 10, the media content is not encrypted on media storage device 999. In one embodiment, if the media content is encrypted on computer 210, it can be decrypted on the computer 210. However, home players and/or stand alone media playing devices rarely include a decryption mechanism, and to experience the music on a home machine, the music is conventionally not encrypted.

10 In one embodiment, an additional component of CCM 300 is that the trigger for agent program 304 may be the copyright bit indicator. This means when the copyright indicator bit is detected by CCM 300, the functions of CCM 300 are initiated. Alternatively, in another embodiment, when the copyright bit indicator is not detected, CCM 300 may remain in an un-invoked or idle state. If CCM 300 can detect the
15 copyright bit indicator, CCM 300 can provide the appropriate compliance with regard to copyright restrictions and/or licensing agreements applicable to the media files.

In an alternative embodiment, a trigger control in the table of contents of media storage device 999 includes instructions for triggering autorun protocol 910 of
20 CCM/MSD 900 and can utilize the copyright indicator bit or alternative implementation to trigger the technology. In this manner, CCM 300 can control copyrighted works while public domain material can be experienced and reproduced at a user's discretion. Because autorun can be problematic for media storage device manufacturers, embodiments of CCM/MSD 900 can include alternative autorun
25 programs that perform analogous to autorun.

In another embodiment, CCM 300 can invoke its own proprietary player, (e.g., custom media device 310 as described with reference to Figure 3), thus enabling increased control of copyright restrictions and/or licensing agreements applicable to the media. By invoking custom media device 310, CCM 300 enables user experience of the media while providing protection against unauthorized reproduction of the media disposed on media storage device 999. .

In an alternative embodiment, the media files and the CCM/MSD 900 disposed on a media storage device 999 are encrypted. This implementation is particularly advantageous for demonstration (demo) versions of media files, beta test versions, and the like that may be disposed on media storage device 999. It is noted that the present embodiment is operable in an online environment, meaning that client computer system 210 is communicatively coupled with web server 250 and/or content server 251 to enable a user experience of the content on a demo version of media storage device 999. In this implementation, CCM 300 allows for specific plays for specific users, which can be controlled via a network, (e.g., network 1000 of Figure 10), and server 250 and/or 251.

In another embodiment, CCM 300 can be implemented for demo and/or pre-release protection. In this embodiment, CCM 300 utilizes sophisticated encryption technology to encrypt the table of contents and CCM 300 with an associated decrypted key located on client computer system 210. Encrypting CCM 300 can also deter nefarious attempts to reverse engineer CCM 300. Decryption can be performed using an associated decryption key. Alternatively, decryption can be performed by a

proprietary or custom media player application resident on demo media storage device, (e.g., 999).

5 The content of media storage device 999 is encrypted, using various levels of encryption to provide protection levels commensurate with copyright holder's desires and required protection. For example, media storage device 999 is delivered to a user or critic for the purposes of review, the user inserts media storage device 999 into the appropriate storage device reader or connector coupled with the journalist's computer (e.g., 210), and CCM 300 is installed on client system 200 in a manner clandestine to the
10 user. Once installed, CCM 300 initiates a communication session with web server 250/content server 251, where content server 251 can provide authorization for the user to experience the media on media storage device 999.

Accordingly, if the user, to whom demo media storage device 999 had been
15 released, had demo media storage device 999 stolen, or if the user allowed alternative parties to try to experience the content of media storage device 999, the unauthorized party would have to try to crack the encryption keys and the encryption of the actual content of media storage device 999, consuming non-trivial amounts of time.

20 Thus, CCM 300 is able to control which users receive authorization to experience the media of media storage device 999, how many times the user may experience the media, and CCM 300 may also define a period of time until the media may no longer be accessible. This may enable copyright holders to release the content on an authorized media storage device, (e.g., 999), prior to "pirated" copies flooding the market.

25

Accordingly, a demo media storage device 999 may be configured such that a first user may get a copy, a second user may get a copy, and if it is known that the second user will share the demo with a third and a fourth user, then the known users would be enabled to experience the media. Advantageously, by virtue of defining
5 which users can access and experience the media, any unauthorized sharing of the media by one of the authorized users can be readily detected, and further sharing or experiencing of the media may be halted. Additionally, because the authorized user shared the media in an unauthorized manner, in a worse case scenario, criminal charges could be filed against that user.

10 It is noted that placing CCM/MSD 900 on a media storage device, (e.g., 999), so as to enable installation of CCM 300 on client system 210 is one manner in which CCM 300 can be installed on client system 210. An alternative manner in which CCM 300 can be installed on client computer system 210 is through "cross-pollination." For example,
15 webcasters broadcast the media file to the user. The media file has a CCM 300 coupled with the media file, and upon downloading the media file onto client computer system 210, embodiments of the present invention enable the installation of CCM 300 onto client computer system 210. In another manner, CCM 300 is incorporated into and becomes part of an operating system operational on client system 210. Alternatively,
20 laws are passed that mandate the inclusion of CCM 300 on each client computer system 210.

Figure 11 is an exemplary logic/bit path block diagram 1100 of a client computer system, (e.g., 210), configured with a copyright compliance mechanism (CCM) 300 for
25 preventing unauthorized reproduction of copyrighted media according to an

embodiment of the present invention. Copyright compliance mechanism 300 is, in one embodiment, coupled with and operational on client system 210 in any manner similar to that described herein with reference to Figures 4, 5A-5D, 6A, 7A-7C, 9, and 10.

5 Diagram 1100 of Figure 11 includes a media storage device media extraction/creation application 1102 communicatively coupled to operating system input/output subsystem 1104 via wave in line 1121 and wave out line 1138. Operating system input/output subsystem 1104 is coupled with media storage device class driver 1106 via wave in line 1123 and wave out line 1136. Media storage device class driver 1106 is coupled with filter driver 1108 via wave in line 1125 and wave out line 1134. Filter driver 1108 is coupled with media storage device port driver 1110 via wave in line 1127 and wave out line 1132. Filter driver 1108 is shown to include a switch 1111, controlled by CCM 300 via coupling 1160. Media storage device port driver 1110 is coupled with media storage device drive 1112 via wave line in 1129 and wave line out 1130. Media storage device 999, shown to include CCM/MSD 900 is receivable by media storage device drive 1112. Additionally, CCM 300 is coupled with operating system input/output subsystem 1104 via wave in line 1150 and wave out line 1151.

20 In one embodiment, CCM 300 is coupled to and controls selectable switch 1111 in filter driver 1108. Depending upon the copyright restrictions and/or licensing agreements applicable to a media file disposed on media storage device 999, CCM 300 controls whether switch 1111 is open (shown), thus preventing the media file from reaching media extraction/creation application 1102, or closed (not shown) so as to allow reproduction of the protected media file. Media extraction/creation application

1102 can be a "ripping" or "burning" application such as Nero, Roxio, Exact Audio Copy, or other readily available application.

Continuing with Figure 11, media storage device 999 is received by media storage device drive 1112. CCM 300 determines whether media storage device 999 or media disposed thereon is protected by any copyright restrictions and/or licensing agreements, e.g., via detection of a copyright indicator bit. CCM 300 communicates with filter driver 1108 to control switch 1111 accordingly. In the present example, reproducing media storage device 999, and/or the contents thereon, would violate applicable restrictions and/or agreements and therefore switch 1111 is in an open position such that the output path, (e.g., wave-out line 1138) to media extraction/creation application 1102 is effectively blocked thereby preventing unauthorized reproduction of media storage device 999.

It is particularly noted that by virtue of CCM 300 controlling switch 1111, and therefore controlling wave-out line 1138, any incoming copyright protected media disposed on a media storage device 999 can be prevented from being reproduced in an unauthorized manner in accordance with applicable copyright restrictions and/or licensing agreements related to the incoming media.

Advantageously, as new secure or proprietary file formats are developed, CCM 300 can be readily adapted to be functional therewith. Further, CCM/MSD 900 can prevent users from making unauthorized reproductions (e.g., recording, copying, ripping, burning, etc.) of media files. By using kernel level filter drivers (e.g., 1108) and getting to a low enough level within the operating system (OS) on client system 210,

CCM 300 can detect particular applications and when they request media storage device drive 1112 to poll the media file for copying, ripping, etc., and disable the data input path. CCM 300, in this embodiment, deals with the input pathway.

5 In one embodiment, alternative applications that monitor the state of client computer system 210 can enable the autorun functionality of client computer system 210 or alternatively, invoke an automatic mechanism similar to autorun to ensure invocation of CCM 300 for compliance of copyright restrictions and/or licensing agreements applicable to media storage device 999 and/or the copyright protected
10 media disposed thereon.

 In one embodiment, CCM 300 can invoke a proprietary media player from media storage device 999, or activate a proprietary media player resident and operable on client computer system 210, or an alternative authorized media player resident on client
15 computer system 210, in a manner similar to that described herein with reference to Figure 3.

 When media storage device 999 is a multisession device, e.g., a compact disk having a data session and a music session (audio tracks), and it is inserted into or
20 communicatively coupled with media storage device drive 1112 such that its content is accessible, CCM 300 views the contents of the media storage device 999, and in some operating systems the audio tracks will not be displayed. Instead, the data session is shown, as is an autorun file, (e.g., autorun protocol component 910), and upon clicking, invokes a player application. CCM 300 can have a data session and files to which a user
25 may not have access unless a player application is invoked.

In one embodiment, the player application could deposit a monitoring portion (e.g., agent program 304) on client system 210, which in one embodiment may reside on client computer system 210 subsequent to removal or decoupling of media storage device 999 from media storage device drive 1112.

By virtue of content in a multisession media storage device 999, which may not be directly accessible to most player applications, the player application can be invoked which can then install the CCM 300 into client system 210, according to one embodiment of the present invention.

In one embodiment, a proprietary media player application is stored on media storage device 999. However, it may not be automatically invoked. Upon some user intervention, e.g., inserting media storage device 999 into media storage device drive 1112, the media player application is loaded onto client system 210 which has CCM 300 integrated therewith. Thus, CCM 300 is launched regardless of autorun being activated or de-activated, and mandates the user to utilize the proprietary media player application, to experience the content of the media, (e.g., media files), on the media storage device 999.

In an alternative embodiment, client computer system 210 has autorun turned off, wherein it is common for the user to be unable to play a media file unless a proprietary media player application is invoked. Activating the proprietary media player application can initiate an installation of those components of CCM 300 that are bypassed when autorun is not active.

Advantageously, by providing a copyright compliance mechanism, (e.g., 300), which can be easily and readily installed on a client computer system, (e.g., 210) one of more embodiments of the present invention can be implemented to control access to, the delivery of, and the user's experience with media content subject to copyright restrictions and/or licensing agreements, for example, as defined by the DMCA. Additionally, by closely associating a client computer system, (e.g., 210), with the user thereof and the media content received, embodiments of the present invention can provide for accurate royalty recording.

Figure 12 is a block diagram of a usage compliance mechanism 1200, an alternative version of copyright compliance mechanism 300 which is configured to be disposed on a media storage device, (e.g., 999 of Figures 10, 11, 13, 14, and 15) in accordance with an embodiment of the present invention. It is noted that CCM 300 of usage compliance mechanism 1200 is similar to CCM 300 as described herein with reference to Figures 3, 4, 5A-5D, 6A, 7A-7C, 8, 9, 10, and 11. Further, usage compliance mechanism 1200 can be readily updated in accordance with global delivery system 800, in a manner similar to that described herein with reference to Figures 7A-7C.

In one embodiment, usage compliance mechanism 1200 can be disposed on a media storage device, (e.g., 999). Content disposed thereon can, in one embodiment, be demonstration and/or pre-release content. Examples of demonstration and/or pre-release content can include, but is not limited to, audio, video, multimedia, graphics, information, data, software programs, etc. More specifically, demonstration and/or pre-release content can contain, but is not limited to, digital movies or music that may

be distributed to persons in the related media field for review, (e.g., a motion picture academy member for their review of a movie, a record industry critic to review songs that may be released on a new compact disc, etc.). Alternatively, demonstration and/or pre-release content can also contain, but is not limited to, a beta version of a software program, and the like.

Alternatively, the content disposed on media storage device 999 can, in another embodiment, be a commercial release of audio content, video content, software application, etc. Embodiments of the present invention are well suited to be implemented in a commercial environment, e.g., public presentation systems such as those in movie theaters, auditoriums, arenas and the like. Additionally, embodiments of the present invention are readily adaptable to be implemented in commercial distribution points, e.g., audio, video, and/or software retail and/or rental establishments, as well as for pay-per-view and/or pay-per-play implementations.

Further, literary works, documents, graphics such as pictures, painting, drawing, and the like can comprise the content on media storage device 999. It is noted that a nearly endless variety of demonstration, pre-release, and/or commercially released content can be disposed on media storage device 999.

Referring to Figure 12, usage compliance mechanism (UCM) 1200 includes an autorun protocol 910 for invoking installation of components of UCM 1200 on a client computer system, (e.g., 210), in one embodiment of the present invention. Autorun protocol 910 of Figure 12 is analogous to autorun protocol 910 of Figure 9. Also

included in UCM 1200 is a file system filter driver 1220, in one embodiment of the present invention.

File system filter driver 1220 can, in one embodiment, be an upper level and/or lower level filter for the individual bus devices within client computer system 210, e.g., media storage device drive 1112 of Figures 10, 11, 13, 14, and 15. File system filter driver 1220 is enabled to hook onto access to a media storage device drive 1112, (e.g., a CD drive), and intercept data reads associated with accessing the content on media storage device 999.

File system filter driver 1220 includes a decrypter 1221 for providing decryption of encryptions applied to encrypted content, (e.g., encryptions 2351-P applied to encryptions 1351-N of media content 2001-M of Figure 13), in one embodiment of the present invention. Decrypter 1221 can provide dynamic decryption of encryptions applied to encrypted media content on a media storage device 999 as the content, (e.g., 2001-M), is accessed and read by media storage device drive 1112.

Still referring to Figure 12, UCM 1200 also includes a secure media player 1210. Secure media player 1210 can be, in one embodiment, similar to custom media device 310, that is an emulation of the custom media device driver 307, as described herein with reference to Figures 3 and 5B-5D. Alternatively, secure media player 1210 may be an alternative media player having controlling properties analogous to custom media device 310. Secure media player 1210 includes a decrypter 1211 for decrypting encryption applied to each instance of media disposed on a media storage device 999, e.g., encryptions 1351 to 1N applied to media content 2001 to N of Figure 13,

respectively. Secure media player 1210 also includes a watermarker 1212 for watermarking the outgoing data stream. In one embodiment, watermarker 1212 operates concurrent with secure media player 1210 and during player 1210 rendering of the content. For example, watermarker 1212 can attach a serial number, e.g., serial number 1380 of Figure 13, associated with each media storage device 999 onto the outgoing data stream.

Figure 13 is a block diagram of contents and components that may be disposed on a media storage device, (e.g., 999), in accordance with embodiments of the present invention. Device 999 is shown with multiple instances of content, (e.g. media content 2001-M), disposed thereon. Media content 2001-M may be, but are not limited to, movies, audio tracks, software, beta software, documents, literary works, etc. It is noted that any digital media can be disposed on media storage device 999 or on a plurality of media storage devices 999.

Media storage device 999 of Figure 13 is analogous to media storage device 999 of Figures 10, 11, 14, and 15. In one embodiment of the present invention, media storage device 999 is configured for utilization in conjunction with demonstration and/or pre-release content.

Media storage device 999 of Figure 13 is shown to have disposed thereon a UCM (usage compliance mechanism) 1200 for controlling presentation of content, (e.g., media content 2001-M), disposed on media storage device 999. The UCM 1200 described herein with reference to Figures 13, 14, and 15, is analogous to the UCM 1200 described

herein with reference to Figure 12. It is noted that autorun protocol 910 of UCM 1200 is, in one embodiment, disposed on media storage device 999 in a non-encrypted format.

Also shown on media storage device 999 is a unique identifier 1380, (e.g., a serial number), for providing a unique identification of the media storage device, in one embodiment of the present invention. Unique identifier 1380 may be, but is not limited to, nearly any distinguishable identifying type of indicator, (e.g., a randomly generated number, a sequential number, a combination of numbers and alphanumeric characters, and the like).

Advantageously, by disposing unique identifier 1380 on a media storage device 999, it enables close association of the content disposed thereon, (e.g., media content 2001-M), with the anticipated recipient, (e.g., a movie critic, a music critic, an academy award member, a software beta tester, etc.), of the media storage device. Therefore, by closely associating a media storage device, (e.g., 999), with an anticipated recipient, (e.g., the user of computer system 210), embodiments of the present invention can prevent unauthorized persons from experiencing content on a media storage device, as described herein with reference to Figures 3, 4, 7A-7C, and 8.

Further advantageous is that by having unique identifier 1380 for each media storage device 999, embodiments also provide security at the media storage device mastering level. This means that an employee working at a mastering facility who dishonestly and/or unlawfully purloins a copy of the media storage device is prevented from copying the contents and turning it into bootleg (unauthorized versions) copies of the media storage device in an attempt to flood the market. Specifically, by virtue of

each media storage device 999 having unique identifier 1380, and each media storage device 999 is associated with its intended recipient, persons not associated with a particular media storage device 999 are unable to experience the content thereon. It is noted that while the market may still be flooded with bootleg copies, those that acquire
5 a bootleg copy of a media storage device 999, in accordance with the present invention, will not be able to experience the content thereon, thereby possibly causing the public to be less receptive to the idea of an inexpensive bootleg copy of something that they cannot use.

10 In one embodiment, media storage device 999 may be distributed to its intended recipients in a variety of ways. For example, distribution of media storage device 999 to its intended recipients can include, but is not limited to, postal delivery methods, e.g., the United States Postal Service, parcel delivery services such UPS (United Parcel Service) and/or Federal Express, courier delivery services, and the like. In another
15 embodiment, the intended recipient of a media storage device 999 may be required to physically pick up device 999 from a distribution point.

Media storage device 999 can include multiple instances of content, e.g., media content 2001-M, in one embodiment of the present invention. Media content 2001-M
20 can be any type of digital media content, including, but not limited to, audio, video, multimedia, graphics, information, data, software programs, etc.

Still referring to Figure 13, in one embodiment of the present invention, each instance of media 2001-M is subject to a first encryption, e.g., encryptions 1351-N,
25 respectively. In one embodiment, a first decryption key for each encryption, e.g.,

encryptions 1351-N, may be stored in a server, (e.g., web server 250 and/or content server 251 of Figures 2, 4, 10, and 14). In one embodiment, secure media player 1210 can utilize decrypter 1211 and the decryption key stored on web server 250 and/or content server 251 to decrypt encryptions 1351-N during rendering of the content. It is noted that secure media player 1210 can be communicatively coupled with web server 250 and/or content server 251 during rendering and presentation of the content disposed on media storage device 999.

Additionally, media content 2001-M having a first encryption applied thereto, e.g., encryptions 1351-N, can each be subject to a second encryption, e.g., encryptions 2351-P, respectively, prior to disposal of media content 2001-N on media storage device 999. In one embodiment, a second decryption key to decrypt encryptions 2351-P may be stored in a server, (e.g., web server 250 and/or content server 251 of Figures 2, 4, 10, and 14). In one embodiment, file system filter driver 1220 can utilize decrypter 1221 and the second decryption key stored on web server 250 and/or content server 251 to decrypt encryptions 2351-P during the reading of the content on media storage device 999 by media storage device drive 1112. It is noted that file system filter driver 1220 can be communicatively coupled with web server 250 and/or content server 251 during rendering and presentation.

In one embodiment, encryptions 1351-N can be less computationally intensive encryptions when compared to encryptions 2351-P. Alternatively, in one embodiment, encryptions 1351-N can be more computationally intensive when compared to than encryptions 2351-P.

There are many available encryption methods that can be implemented as encryptions 1351-N and/or encryptions 2351-P. Examples of encryptions that may be implemented for encryptions 1351-N and/or 2351-P can include, but are not limited to, triple DES (data encryption standard), AES (advanced encryption standard), Blowfish, and numerous others. In one embodiment, encryptions 1351-N and/or 2351-P can each be comprised of a series and/or a mixture of encryptions. A differing encryption, e.g., a plurality of randomly generated encryptions, can be implemented for each instance of media on a media storage device, rather than using one format. In one embodiment, numerous variations of Blowfish are utilized to provide the desired encryptions.

It is noted that when the media (e.g., 2001) of media storage device 999 is encrypted utilizing multiple different encryptions (e.g., 1351 and 2351), the media is more secure against those with dishonest and/or unlawful interests. For example, a person/hacker may attempt to gain access to the content by breaking the second encryption (e.g., 2351) applied to a media content (e.g., 2001). However, if they are successful, the remaining encryption, (e.g., 1351) remains unbroken by virtue of the differing encryptions. Therefore, the person/hacker would have to perform the entire encryption breaking process again to access media content 2001 on media storage device 999. Thus, after spending non-trivial amounts of time breaking two differing encryptions applied to an instance of media (e.g., 2001) the remaining content on media storage device 999 can still be encrypted, each with its own differing multiple encryptions.

Figure 14 is a block diagram of a communicative environment 1400 for controlling presentation of media content disposed on a media storage device. Included

in communicative environment 1400 is a media storage device drive 1112 coupled with a client computer system 210 via a data/address bus 110. Client computer system 210 is coupled with web server 250 and/or content server 251 via Internet 201. A media storage device 999, upon which a usage compliance mechanism 1200 may be disposed, is received by in media storage device drive 1112. Autorun protocol component 910 detects the reception and automatically invokes installation of CCM 300, file system filter driver 1220, and secure media player 1210 from media storage device 999 into client computer system 210. Subsequent to installation, UCM 1200 initiates a dynamic update with web server 250 and/or content server 251, via Internet 201, as described herein with reference to Figures 3, 4, and 7A-7C, thereby controlling the integrity of the software. Additionally, by conferring with servers 250 and/or 251 via Internet 201 online, the UCM 1200 software version on media storage device 999 and installed on client computer system 210 can be updated when circumventions occur and kept current from platform to platform.

Advantageously, the monitoring mechanism of agent program 304 enables constant morphing of the version of CCM 300 disposed on media storage device 999 by communicating with server 250 and/or 251 and utilizing the dynamic update capabilities of global network 800 to readily update that which has been installed on client computer system 210, via media storage device 999.

In one embodiment, the installation is performed clandestine with respect to the recipient of media storage device 999 and is initiated by inserting media storage device 999 into an appropriate media storage device drive, (e.g. a magnetic/optical disk drive or alternative device drive) coupled with client system 210. Portions of UCM 1200

determine if the recipient is registered with web server 250 and/or content server 251. If the recipient is not registered with servers 250 and/or 251, as described herein with reference to Figures 4, 7A-7C, and 8, portions of UCM 1200 initiates an installation process as described herein with reference to Figures 3, 4, 7A-7C, and 11.

5

If computer system 210 is registered with servers 250 and/or 251, UCM 1200 can initiate an update process with web server 250 and/or content server 251 to readily include updates that have been invoked subsequent to distribution of media storage device 999. By virtue of the dynamic update capabilities of UCM 300, regardless of the version of CCM 300 on media storage device 999, UCM 1200 provides compliance with copyright restrictions and/or licensing agreements applicable to the media content on media storage device 999, (e.g., media content 2001-M). Advantageously, enabling dynamic adaptability of UCM 1200 provides for continued interoperability with new and updated operating systems, advancements in electronic technology, communication technologies and protocols, and the like, ensuring the effectiveness of UCM 1200 into the future.

In another embodiment, if the user is a registered user with global delivery system 800, UCM 1200 can detect which version is most current. Accordingly, when the version existing on client system 210 is more current than the version (for install) on media storage device 999, UCM 1200 can bypass the install process and present the contents contained on media storage device 999 to the user for them to experience.

Further advantageous, this technology is backward compatible with media storage device drives manufactured subsequent to 1982. Additionally, UCM 1200 is

compatible with media storage devices having a copyright indicator bit disposed thereon. The copyright indicator bit has been included on all CDs released since 1982.

In the present embodiment of Figure 14, each instance of media is encrypted on media storage device 999, as described herein with reference to Figure 13. However, most home players and/or stand alone media playing devices rarely include a decryption mechanism. As such, to experience the music on a home machine, the music is conventionally not encrypted. Accordingly, media storage device 999, in its present embodiment, may not be operable on a home and/or stand alone media playing device.

In one embodiment, UCM 1200 can invoke its own proprietary player, (e.g., secure media player 1210), as described with reference to custom media device 310 of Figure 3, thus enabling increased control of copyright restrictions and/or licensing agreements applicable to the media content. By invoking a secure media player 1210, UCM 1200 enables user experience of media content while providing protection against unauthorized presentation or reproduction of the media disposed on media storage device 999.

Still referring to Figure 14, in one embodiment, the media content, (e.g., media content 2001-M), and UCM 1200 disposed on a media storage device 999 are encrypted, with the exception of autorun protocol 910, as described above. In one embodiment of the present invention, UCM 1200 is encrypted differently than media content 2001-M, thereby preventing the cracking of one encryption from being utilized on another encryption. This implementation is particularly advantageous for demonstration (demo) versions of media files, beta test versions, and the like that may be disposed on

media storage device 999. It is noted that the present embodiment is operable in an online environment, meaning that client computer system 210 can be communicatively coupled with web server 250 and/or content server 251 to enable a user experience of the content on a demo version of media storage device 999. In this implementation, UCM 1200 allows for specific plays for specific users, which can be controlled via a network, (e.g., network 1400), and server 250 and/or 251.

In the present embodiment, UCM 1200 can be implemented for demonstration and/or pre-release protection of content disposed on a media storage device 999.

However, content disposed on media storage device 999 can also be commercially released content, (e.g., audio, video, software, and the like). In this embodiment, sophisticated encryption technology, (e.g., Blowfish), is utilized to encrypt media content 2001-M on media storage device 999 with an associated decrypter key located on web server 250 and/or content server 251. In one embodiment, a plurality of encryptions are applied to media content 2001-M and a plurality of decrypter keys are stored on web server 250 and/or content server 251. Decryption can be performed using an associated decryption key in conjunction with a secure media player 1210 and file system filter driver 1220 installed on computer system 210 via media storage device 999.

Still with reference to Figure 14, the content (e.g., media content 2001-M) of media storage device 999, is encrypted using various levels of encryption to provide protection levels commensurate with copyright holders desires and required protection. For example, media storage device 999 is delivered to a user or critic for the purposes of review. The user inserts media storage device 999 into the appropriate storage device

reader or connector coupled with the recipient's computer, and autorun protocol 910 initiates UCM 1200 install of CCM 300, file system filter driver 1220, secure media player 1210 on client system 210 in a manner clandestine to the user. Once installed, UCM 1200 initiates a communication session with web server 250/content server 251, 5 where content server 251 can provide authorization for the user to experience the media on media storage device 999.

Accordingly, if the user, to whom demo media storage device 999 had been released, had demo media storage device 999 stolen, or if the user allowed alternative 10 parties to try to experience the content of media storage device 999, the unauthorized party would have to try to crack the encryption keys and the encryption of the actual content of media storage device 999, consuming non-trivial amounts of time.

Thus, UCM 1200 is able to control which recipients receive authorization to 15 experience the media content on media storage device 999, how many times the recipient may experience the media, and UCM 1200 may also define a period of time beyond which the media content may no longer be accessible. This may enable copyright holders to release the media content on an authorized media storage device, (e.g., 999), prior to pirated copies flooding the market.

20 Still referring to Figure 14, accordingly, a media storage device 999 may be configured such that a first user may get a copy, a second user may get a copy, and if it is known that the second user will share the demo with a third and a fourth user, then the known users would be enabled to experience the media. Advantageously, by virtue 25 of defining which users can access and experience the media, any unauthorized sharing

of the media by one of the authorized users can be readily detected, and further sharing or experiencing of the media may be halted. Additionally, since the authorized user shared the media in an unauthorized manner, in a worse case scenario, criminal charges could be filed against that user.

5

It is noted that by including placing UCM 1200 on a media storage device, (e.g., 999), so as to enable installation of CCM 300 on client system 210 is one manner in which CCM 300 can be installed on client system 210. An alternative manner in which CCM 300 can be installed on client computer system 210 is through "cross-pollination."

10 For example, webcasters broadcast the media file to the user. The media file has a CCM 300 coupled with the media file, and upon downloading the media file onto client computer system 210, embodiments of the present invention enable the installation of CCM 300 onto client computer system 210. In another manner, CCM 300 is incorporated into and becomes part of an operating system operational on client system
15 210. Alternatively, laws are passed that mandate the inclusion of CCM 300 on each client computer system 210.

Figure 15 is an exemplary logic/bit path block diagram 1500 of a client computer system, (e.g., 210), configured with a usage compliance mechanism (e.g., 1200) for
20 controlling presentation of content on a media storage device (e.g., 999), in accordance with one embodiment of the present invention. It is noted that usage compliance mechanism 1200 of Figure 15 is analogous to usage compliance mechanism 1200 of Figure 12. Therefore, CCM 300 of usage compliance mechanism 1200 is analogous to a copyright compliance mechanism 300 coupled with and installed on a client computer

system, (e.g., 210), as described herein with reference to Figures 3, 4, 5A-5D, 6A, 7A-7C, 8, 9, 10, 11 14, 15, and 16.

Diagram 1500 includes a media storage device drive (e.g., 1112) coupled with a
5 media storage device file system driver 1114 via line 1571. Media storage device drive
file system driver 1114 enables an operating system, (e.g., Windows by Microsoft,
Apple, Linux, etc.), on a client computer system, (e.g., 210), to recognize and control the
media storage device drive 1112. Coupled to media storage device drive file system
driver 1114 is file system filter driver 1220, via line 1572. Coupled to file system filter
10 driver 1220 is a secure media player 1210 via line 1573. Coupled with secure media
player 1210 are an operating system media subsystem 503 via line 1577 and a media
hardware output device 1370 via line 1574. UCM 1200 is coupled with operating
system media subsystem 503 via line 1576.

15 Media storage device drive 1112 of Figure 15 is analogous to media storage
device drive 1112 of Figures 11 and 14. Media storage device drive 1112 is configured
to receive a media storage device 999. In one embodiment, drive 1112 may be a CD
drive and media storage device 999 would be a CD. In another embodiment, drive 1112
may be a DVD drive and accordingly, media storage device 999 would be a DVD, and
20 so on. Therefore, media storage device drive 1112 can, when so configured, receive any
media storage device 999 upon which data or content may be disposed.

File system filter driver 1220 can be an upper level and/or lower level filter for
individual bus devices, (e.g., media storage device drive 1112), within client computer
25 system 210, and is analogous to file system filter driver 1220 of Figure 12. File system

filter driver 1220 is able to hook onto access of media storage device drive 1112, (e.g., a CD drive) and intercept data reads associated with accessing the content, (e.g., media content 2001-M), of media storage device 999. File system filter driver 1220 is also enabled, via decrypter 1221 and a decrypter key on servers 250 and/or 251, to provide
5 dynamic decryption of encrypted media content on media storage device 999 as the content is accessed and read by media storage device drive 1112.

By virtue of file system filter driver 1220 operating at a file system level instead of operating at a device drive class level, (e.g., a CD class level), it is able to recognize
10 which files are being accessed from media storage device 999 for a particular operation. Advantageously, this obviates the need for a file system to be implemented within a driver for determining whether data that is being read needs decrypting.

Still referring to Figure 15, secure media player 1210 is analogous to secure
15 media player 1210 of Figure 11. Secure media player 1210 can, in one embodiment, be a custom media device 310 emulated by a custom media device driver 307, as described herein with reference to Figure 3. In another embodiment, secure media player 1210 can be a proprietary player configured for utilization with demonstration and/or pre-release content disposed, (e.g., 2001-M), on a media storage device, (e.g., 999). Other
20 authorized media players may also be used to present media content of a media storage device 999, provided the other media players can comply with usage restrictions and/or licensing agreements applicable to the media content and provided by secure media player 1210.

Media hardware output device 1370 is an appropriate output device for the media content of media storage device 999. If media content 2001-M are audio tracks or songs, then output device 1370 is an audio or sound card for outputting music via speakers. Alternatively, if media content 2001-M are video tracks, movies, literary works, software programs, etc., then output device 1370 is a graphics card for outputting movies, text, and the like via a display device, (e.g., 105 of Figure 1).

Continuing with Figure 15, media storage device 999 is received by media storage device drive 1112. Autorun protocol 910 initiates a process to determine the presence of a usage compliance mechanism 1200 and a secure media player, (e.g., 1210), operable on computer system 210. If either and/or both usage compliance mechanism 1200 and secure media player 1210 are not present on computer system 210, autorun protocol 910 initiates installation of the components, as described herein with reference to Figures 3, 4, 5A-5D, 6, 7A-7C, and 8-16. If UCM 1200 and secure media player 1210 are both present, autorun 910 bypasses the installation thereof. Media storage device file system driver 1114 accesses the content, (e.g., 2001-M), on media storage device 999 and reads the data.

File system filter driver 1220 intercepts the read operation being performed by driver 1114 and dynamically decrypts a second encryption, (e.g., 2351-P), applied to media content 2001-M via decrypter 1221 and a second decryption key stored on and retrieved from servers 250 and/or 251. In one embodiment, if file system filter driver 1220 is not communicatively coupled with server 250 and/or 251, thereby enabling retrieval of the second decryption key, presentation of the content of media storage device 999 is not permitted.

Continuing with Figure 15, subsequent to the second encryptions 2351-P being decrypted, media content 2001-M, which is still encrypted with a first encryption, (e.g., 1351-N, respectively), is output to secure media player 1210 via line 1573. Secure media
5 player 1210 in conjunction with UCM 1200 communicates with server 250 and/or 251 and determines if computer system 210 and the user thereof, are authorized to experience media content 2001-M. If system 210 and the user thereof are authorized to experience media content 2001-M, secure media player 1210 commences to render the media content for presentation via media hardware output device 1370.

10 Concurrent with rendering media content 2001, secure media player 1210 can, in one embodiment, communicate with server 250 and/or 251 and retrieve the decryption key associated with each encryption, (e.g., 1351-N), and with decrypter 1211 of Figure 12, in order to dynamically decrypt each instance of media, (e.g., 2001-M), as the content
15 is being rendered and output via line 1574 to media hardware output device 1370.

Because rendered content is vulnerable to capture and/or imaging, and thus becoming subject to ripping, burning, copying, and the like, secure media player 1210 can watermark, (e.g., via watermarker 1212), the outgoing data stream that is output to
20 media hardware output device 1370 via line 1574. In one embodiment, utilizing watermarker 1212, the outgoing data stream is watermarked concurrent with the rendering performed by secure media player 1210. Further, secure media player 1210 can attach a unique identifier, (e.g., 1380), with each rendered media content (e.g., 2001-M). In one embodiment, the unique identifier 1380 is a serial number that is attached to
25 each media content, (e.g., 2001-M), as it is being rendered and output to media

hardware output device 1370. In this manner, if the rendered content being output is somehow captured, imaged, etc., by virtue of the association of unique identifier 1380, (e.g., serial number), with media storage device 999 and the media content disposed thereon, (e.g., content 2001-M), and computer system 210 with which the recipient of
5 media storage device 999 is associated, unauthorized presentation and reproduction of the media content is prevented.

Figure 16 is a flowchart 1600 of computer implemented steps performed in accordance with one embodiment of the present invention for controlling presentation
10 of media content disposed on a media storage device. Flowchart 1600 includes processes of the present invention, which, in some embodiments, are carried out by processors and electrical components under control of computer readable and computer executable instructions. The computer readable and computer executable instructions reside, for example, in data storage features such as computer usable volatile memory
15 104 and/or computer usable non-volatile memory 103 of Figure 1. However, the computer readable and computer executable instructions may reside in any type of computer readable medium. Although specific steps are disclosed in flowchart 1600, such steps are exemplary. That is, the present embodiment is well suited to performing various other steps or variations of the steps recited in Figure 16. Within the present
20 embodiment, it should be appreciated that the steps of flowchart 1600 may be performed by software, by firmware, by hardware or by any combination thereof .

It is noted that flowchart 1600 is described in conjunction with Figures 2, 3, 4, 5A-5D, 6, 7A-7C, and 8-15 to more fully describe the operation of the present embodiment.

25 At operation 1610, an autorun mechanism, (e.g., autorun protocol 910), disposed on a

media storage device (e.g., 999), is activated in response to a computer system, (e.g., 210), receiving the media storage device in an appropriate device drive, (e.g., media storage device drive 1112).

5 At operation 1612 of Figure 16, a monitoring program disposed on media storage device 999 determines if a usage compliance mechanism, (e.g., UCM 1200), is installed on the computer system (e.g., 210) which received media storage device 999. In one embodiment, agent programs 304 may perform the determination at operation 1612. However, in another embodiment, combinations of components of a CCM 300, as
10 described herein with reference to Figure 3, 4, 7A-7C may be utilized to perform the determination at operation 1612. If a usage compliance mechanism is not present on computer system 210 at operation 1612, process 1600 proceeds to operation 1611. Alternatively, if a usage compliance mechanism is present on computer system 210 at operation 1612, the process 1600 proceeds to operation 1614.

15

 At operation 1611, a usage compliance mechanism can be installed on computer system 210. It is noted that the installation of the usage compliance mechanism on computer system 210 at operation 1611 may be performed in a wide variety of ways in accordance with the present embodiment. For example, the installation can be
20 implemented at operation 1611 as described herein with reference to Figures 3, 4, 5A-5D, 6, 7A-7C, and 8-15, but is not limited to such.

 At operation 1614, a monitoring mechanism (e.g., of UCM 1200) disposed on media storage device 999 determines if a secure media player, (e.g., 1210) is present and
25 operable on computer system 210. In one embodiment, agent program 304 of CCM 300

of Figure 3 can perform the determination at operation 1611. If a secure media player, (e.g., 1210) is not present and operable on computer system 210 at operation 1614, the present method proceeds to operation 1611. Alternatively, if a secure media player (e.g., 1210) is present and operable on computer system 210 at operation 1614, the
5 present method proceeds to operation 1616.

At operation 1611 of figure 16, a secure media player (e.g., 1210) can be installed on computer system 210. It is noted that the installation of the secure media player on computer system 210 at operation 1611 may be performed in diverse ways in
10 accordance with the present embodiment. For example, the installation can be implemented at operation 1611 as described herein with reference to Figures 3, 4, 5A-5D, 6, 7A-7C, and 8-15 but is not limited to such.

At operation 1613, a determination is made as to whether the installation (e.g., of
15 a usage compliance mechanism and/or a secure media player) at operation 1611 was successful. If so, process 1600 proceeds to operation 1616. However, if it was determined at operation 1613 that the installation at operation 1611 was not successful, process 1600 proceeds to operation 1616.

20 At operation 1616, a determination is made as to whether computer system 210 and the user thereof are authorized to experience media content (e.g., 2001-M) of media storage device 999. It is noted that the determination at operation 1616 may be performed in a wide variety of ways in accordance with the present embodiment. For example, the usage compliance mechanism (e.g., 1200) can communicate with servers
25 250 and/or 251 in networks 200, 400, 1000 and/or 1400 to determine whether computer

system 210 and its user are authorized to experience media content (e.g., 2001-M) on media storage device 999, as described herein with reference to Figures 3, 4, operations 704-708 of Figure 7A, and Figures 8-15, but is not limited to such. If computer system 210 and its user are not authorized to experience the content on media storage device 999 at operation 1616, process 1600 proceeds to operation 1615. Alternatively, if computer system 210 and its user are authorized to experience the content on media storage device 999 at operation 1616, process 1600 proceeds to operation 1618.

At operation 1618 of Figure 16, a determination is made (e.g., by UCM 1200) as to whether secure media player 1210, usage compliance mechanism 1200 and computer system 210 are all communicatively coupled with a network (e.g., 200, 400, 800, 1000, and/or 1400) in accordance with the present embodiment. If one or more of the conditions are not met at operation 1618, the present method proceeds to operation 1615. Alternatively, if secure media player 1210, usage compliance mechanism 1200 and computer system 210 are all communicatively coupled with the network (e.g., 200, 400, 800, 1000, and/or 1400) at operation 1618, the present method proceeds to operation 1620.

At operation 1615, the presentation of content (e.g., 2001-M) of media storage device 999 to the user of computer system 210 is prevented. Alternatively, computer system 210 and the user thereof may communicate with the network (e.g., 200, 400, 1000, and/or 1400) and attempt to establish credentials and/or to re-establish a communicative coupling with the network that would allow presentation of the content, as described herein with reference to Figure 3, 4, and steps 704-708 of Figure 7A.

At operation 1622, the session is ended.

At operation 1620 of Figure 16, media content (e.g., 2001-M) of media storage
5 device 999 can be presented to the user of computer system 210. It is noted that
operation 1620 can be implemented in a wide variety of ways in accordance with the
present embodiment. For example, the media content (e.g., 2001-M) on media storage
device 999 can be read by media storage device drive 1112. File system filter driver
1220 can intercepts the read operation being performed by media storage device file
10 system driver 1114 and dynamically decrypts a second encryption (e.g., 2351-P) applied
to media content 2001-M via decrypter 1221 and a second decryption key stored on and
retrieved from servers 250 and/or 251. Within an embodiment, if file system filter
driver 1220 is not communicatively coupled with server 250 and/or 251, enabling
retrieval of the second decryption key, presentation of the content on a media storage
15 device 999 is not permitted.

Continuing with an exemplary implementation of operation 1620, subsequent to
a second encryption (e.g., 2351-P) being decrypted, media content 2001-M, which is still
encrypted with a first encryption, (e.g., 1351-N, respectively), is output to secure media
20 player 1210. In one embodiment, secure media player 1210, in conjunction with UCM
1200, communicates with server 250 and/or 251 and commences to render the media
content for presentation via a media hardware output device, (e.g., 1370).

It is noted that concurrent with rendering media content, (e.g., 2001-M), secure
25 media player 1210, can, in one embodiment, communicate with server 250 and/or 251

and retrieve the decryption key associated with each encryption, (e.g., 1351-N), and with decrypter 1211 of Figure 12, dynamically decrypt each instance of media, (e.g., 2001-M) as the content is being rendered and output the rendered content to media hardware output device 1370.

5

Since rendered content is vulnerable to capture and/or imaging, and thus becoming subject to ripping, burning, copying, and the like, secure media player 1210 can watermark, (e.g., via watermarker 1212), the outgoing data stream that is output to media hardware output device 1370. In one embodiment, the outgoing data stream is
10 watermarked concurrent with the rendering performed by secure media player 1210. Further, secure media player 1210 attaches a unique identifier (e.g., 1380) with each rendered media content 2001-M. In one embodiment, a unique identifier 1380 (e.g., a serial number) is attached to each media content (e.g., 2001-M), as it is being rendered and output to media hardware output device 1370. In this manner, if the rendered
15 content being output is somehow captured, imaged, etc., by virtue of the association of unique identifier 1380 with media storage device 999 and the media content (e.g., 2001-M) disposed thereon and the computer system 210 with which the recipient of media storage device 999 is associated, unauthorized presentation and reproduction of the media content is prevented.

20

In another implementation, embodiments of the present invention can be utilized in a distributed network topology to control media sharing among computer system within the network. Some of the distributed network topologies in which embodiments of the present invention can be utilized are, but is not limited to, a centralized, a ring, a
25 hierarchical, and a decentralized distributed topology. A decentralized distributed

network topology is commonly referred to as a peer-to-peer (P2P) network. In a P2P network, each computer system/node in the network is able to communicate with any other computer system/node within the network. A computer system in a P2P network can request data from the network as well as provide data to the network. Thus, a
5 computer system can be both a requesting computer system (requesting node) and a source computer system (source node) within a P2P network. Additionally, there may be a host server or a plurality of host servers present in the network that may be utilized alone or in combination with each other (e.g. as an administrative node, supernode, etc.) to provide administrative functionalities to the computer systems therewithin and for
10 providing content to the P2P network.

It is known that many implementations of a distributed network topology, (e.g., a decentralized topology), are not configured to adequately control and/or monitor the exchanging or trading of media among computer systems in the P2P network.

15 Embodiments of the present invention provide a novel and inventive solution to this problem.

Figure 17 is a block diagram of a network environment 1700 for sharing media content between nodes (e.g., computer systems) communicatively coupled therewith, in
20 accordance with one embodiment of the present invention. Network 1700 includes a client node 1705, a source node 1715 and an administrative node 1770 which are communicatively coupled via communication link 1710. Communication link 1710 may be a wireline, wireless, or combination of wireline and wireless technologies and communication protocols that facilitate interaction between computer systems.

25 Additionally, in the present embodiment, nodes 1705, 1715, and 1770 may each

implemented in a similar manner to that described herein with reference to Figure 1. Further, network environment 1700 may be implemented in a manner similar to that described herein with reference to network 200 of Figure 2 and network 800 of Figure 8.

5 In one embodiment, client node 1705 may be implemented in a manner similar to a client computer system 210 as described herein with reference to Figures 2, 4-8, 10, 11, 14, and 15. Source node 1715 may, in one embodiment, be a client computer system 1705 that makes available to network 1700 media content that may be stored therewithin. In another embodiment, source node 1705 may be a media distribution
10 point (MDP) as described herein with reference to Figure 8, configured to make available to network 1700 media content that may be stored therewithin. By virtue of the functionality inherent to a P2P network, source node 1715 can function as a client node while client node 1705 can function as a source node.

15 Still referring to Figure 17, administrative node 1770 may, in one embodiment, be implemented in a manner similar to a web server 250 and/or content server 251, as described herein with reference to Figures 2 and 4. As such, administrative node 1770 may also be coupled to a database, (e.g., database 451 and/or 451 of Figure 4). Accordingly, administrative node 1770 can, in one embodiment, be configured to
20 provide management functionalities to a network, (e.g., network 1700 or network 800 of Figure 8). Types of management functionalities that can be provided by administrative node 1770 can include, but is not limited to, network management, user management, encryption and decryption key management, authorization management, media management, transaction management, player application management, and cache

management. Administrative node 1770 can also be implemented as a source node 1715 and/or as a media delivery point as described herein with reference to Figure 8.

Network management can include, but is not limited to, determining the route through which an instance of media is transferred to another computer system in network 1700. For example, and referring to Figure 8, an instance of media is located in a plurality of locations within network 800, e.g., media delivery points 808 (San Jose), 816 (Tokyo), and 812 (New York City). Client node 1705 located in Washington D.C. requests that particular instance of media content. Administrative node 1770 is able to determine, for this example, that the least amount of network resources needed to transfer the media content to requesting client node 1705 would be transferring the instance of media content from media delivery point 812 (New York City).

It is anticipated that many instances of a particular instance of media may be disposed throughout the network, (e.g., 1700), in which there may be varying degrees of quality among the instances media content. Further, it is known that differing source nodes (e.g., 1715) can have varying rates of transfer. For example, a source node 1715 may be coupled to network 1700 via a digital subscriber line (DSL), whereas another source node 1715 may be coupled to network 1700 via a 56K modem, which is substantially slower than a DSL connection. Additionally, transfer routes that would go through a network connection that may be experiencing downtime or technical difficulties can be rerouted. Advantageously, administrative node 1770 can incorporate quality of media content, connection health, and/or delivery speed information into its determination from which source node 1715 the instance of media is to be delivered.

Still referring to Figure 17, user management, in one embodiment, can include, but is not limited to, maintaining and verifying current user information such as user name, password, billing address, valid credit card number, valid online payment or alternative electronic payment service, MAC (media access control) address, etc., as
5 described herein with reference to Figures 3 and 4, and steps 704 - 710 of Figures 7A-7C.

In one embodiment, encryption and decryption key management includes, but is not limited to, storing an encryption key associated with a client computer system for use by the client computer system to encrypt media content thereon into an encryption
10 local to that client computer system, managing a plurality of intermediate encryption and decryption keys for utilization with media content that is being transmitted to a client computer system, etc.

In one embodiment, encryption and decryption key management includes, but is
15 not limited to, storing an encryption key associated with a client node (e.g., 1705) for use by the client node to encrypt media content thereon into an encryption local to that client node, managing a plurality of intermediate encryption and decryption keys for utilization with media content that is being transmitted to a client node, etc.

20 In one embodiment, authorization management may include verifying that a client node 1705 is authorized to be connected to network 1700, thus being associated with the network. Alternatively, authorization management may also include, but is not limited to, authorizing sharing of an instance of media with network 1700 that does comply with copyright restrictions and/or licensing agreements applicable to the
25 particular instance of media. Furthermore, authorization management may include, but

is not limited to, prohibiting sharing of an instance of media with network 1700 that does not comply with copyright restrictions and/or licensing agreements applicable to the particular instance of media.

5

Still referring to Figure 17, media management can include, but is not limited to, storing a list of available instances of media content and its associated location within network 1700, storing information related to each instance of media, (e.g., title, artist, genre, length/duration of media content, bit pattern information related to a particular instance of media for identification, storing information regarding royalty fees (where applicable) that may be assessed to an instance of media content each time it is transmitted within the network), storing information to enable expiring the media content on a computer system when the computer system is no longer authorized to access and/or is no longer associated with network 1700, and the like.

15

Additionally, media management can also include utilizing an Ethernet filter and other mechanisms to obtain information related to transmitted and/or received instances of media, instances of media that have recently entered the network, and the like. In one embodiment, an Ethernet filter may be coupled to a network interface device (e.g., 104) coupled to the computer system receiving the instance of media content. It is noted that the network interface may include, but is not limited to, a modem, a NIC (network interface card), a wireless receiver, etc. Information obtained by an Ethernet filter can be used in identifying a particular instance of media, determining appropriate royalty fees, etc.

25

In one embodiment, transaction management can include generating a transaction for each successful transfer of media content from one node (e.g., source 1715) to another node (e.g., client 1705) in the network, e.g., from source computer system 1715 to client computer system 1705 in the network (e.g., 1700). In one
5 embodiment, the receiving client node 1705 acknowledges successful receipt of the media content and accordingly, administrative node 1770 generates a transaction applicable to the receiving client node 1705 and which is associated with that particular instance of media. Since different instances of media content may have differing usage and/or royalty fees associated therewith, embodiments of the present invention can
10 provide the mechanisms to account for each transfer such as generating a transaction for each transfer, generating a payment from the recipient of the instance of media, and/or delivering an appropriate remuneration to the media content copyright holder.

Advantageously, a royalty transaction can also be applied to a copyrighted
15 instance of media where no royalty payment was previously possible. For example, assume that a bootleg copy of an artist's live performance has been freely available for years. This bootleg copy may have been made from a portable recorder brought into a live performance and was used to record the performance. In this example, a person acquires an analog bootleg copy and converts it to a digital format or acquires a digital
20 format of the bootleg copy. The digital copy is then availed to network 1700 for sharing among the nodes coupled therewith. It is noted that when an instance of media enters network 1700, it becomes protected as described herein with reference to Figures 3, 4-9, 11, 12, and 15.

Accordingly, when a client node 1705 successfully receives the digital copy, a transaction is generated, thus invoking a royalty payment when no such royalty payment was previously possible. Advantageously, this can provide a means to distribute appropriate remuneration to copyright holders and licensees that would otherwise receive no payment.

Continuing, when an instance of media can be freely distributed, a transaction is still generated. This advantageously provides a record of the numbers of requests for a particular instance of media, the location from where those requests originated, and the like, thereby providing data which can be utilized in marketing studies and implementing marketing strategies, etc.

It is particularly noted that in an alternative implementation, embodiments of the present invention can be readily configured to provide a mechanism that can generate a sales/use tax transaction related to transfer of a instance of media. Embodiments of the present invention can be readily configured to calculate appropriate sales/use tax amounts relative to the location of the receiving client computer system.

Advantageously, this would enable municipalities, counties, states, and other governmental agencies to increase revenue, thus possibly realizing a reduction in a deficit associated with a government. It is anticipated that generating sales/use tax transactions could provide from tens to hundreds of millions of dollars in previously untapped revenue to those governmental agencies.

Still referring to Figure 17, player management, in one embodiment, can include, but is not limited to, maintaining secure player application integrity, adding additional

player applications to a list of approved player applications, removing player applications that do not meet usage restriction requirements applicable to a media file, modification of a secure player application to improve its functionality, modification to counteract nefarious hacking attempts, disabling the secure player application when the computer system on which it is operable is no longer authorized to participate in or is associated with network 1700, etc.

Cache management, in one embodiment, can include, but is not limited to, maintaining protected media container file integrity, modification thereof to improve functionality or to counteract detected nefarious activities, disabling the protected media container file when the computer system upon which the protected media container file is disposed is not authorized to access and/or is no longer associated with network 1700.

In the present embodiment, client node 1705 and source node 1715 each have an instance of a usage compliance mechanism (UCM) 1800 coupled therewith. In the present embodiment, UCM 1800 of Figure 17 and as described with detail in Figure 18 is similar to a copyright compliance mechanism 300 and/or a usage compliance mechanism 900 and/or 1200, as described herein with reference to Figures 3, 4, 5A-5D, 6A, and/or 9, 11, 12, and 15, respectively.

Figure 18 is a block diagram of components in a usage compliance mechanism (UCM) 1800 that is configured to be installed and operable on a client node (e.g., 1705) and/or a source node (e.g., 1715) in accordance with an embodiment of the present invention. UCM 1800 includes a copyright compliance mechanism (e.g., 300) that can

be analogous to a copyright compliance mechanism 300 as described herein with reference to Figures 3, 4, 5A-5D and 6A. It is noted that UCM 1800 also includes those features and components as provided in a UCM 900 and a UCM 1200 as described herein with reference to Figures 9, 11, 12, and 15, respectively.

5

UCM 1800 further includes a secure player application 1810, a client communication application 1820, and a media storage container creator 1830. Media storage container creator 1830 is configured to allocate a portion of a memory unit coupled to the computer system in which UCM 1800 is installed, (e.g., 102 and/or 103) coupled to the computer system (e.g., 100) into which UCM 1800 is installed. Media storage container creator 1830 utilizes the allocated portion of a memory unit and creates a protected media container file (e.g., a custom file system) into which received and/or availed instances of media, (e.g., audio files, video files, multimedia files, documents, software, and the like) are stored. It is noted that in one embodiment, media content that is stored in a protected media container file (not shown) is, in addition to other encryptions applicable to the instance of media, encrypted local to the computer system on which the protected media file container is disposed. In an example, an instance of media is stored on both client node 1705 and source node 1715 of Figure 17. Accordingly, media content is uniquely encrypted local to node 1705 and is uniquely encrypted local to node 1715. Additionally, in the context of the present invention, the term availed and/or availing refers to making available to a network, (e.g., 1700), an instance of media that may be stored in a protected media container file.

Still referring to Figure 18, UCM 1800 also includes, in one embodiment, a client communication application 1820. Client communication application is a custom client

communication application configured to provide communication functionality between the nodes in a decentralized distributed network, (e.g., 1700). Alternatively, client communication application 1820 is also well suited to be implemented in nearly any network, including, but not limited to, network 1700. Client communication
5 application 1820 can be a well known and readily available communication application and which may be written in a commonly utilized programming language including, but not limited to, C, C++, Java, Fortran, etc.

In one embodiment, client communication application 1820 is configured to
10 decrypt media content from an encryption local to a computer system on which application 1820 is operable. Client communication application 1820 can also be configured to encrypt media content into an intermediate encryption for transferring the media content to another computer system communicatively coupled with a network, (e.g., 1700). Application 1820 is additionally configured to decrypt media
15 content that is received in an intermediate encryption from a computer system coupled with network 1700. Client communication application 1820 is further configured to encrypt the media content into an encryption local to the computer system on which application 1820 is operable.

20 In one embodiment, client communication application 1820 can utilize an intermediate encryption key provided by an administrative node (e.g., 1770) to encrypt the media content into the intermediate encryption for transfer. Application 1820 can also utilize an intermediate decryption key, also provided by administrative node 1770, to decrypt media content that is received in an intermediate encryption from a
25 computer system coupled with network 1700.

In one embodiment, client communication application 1820 can transmit an indicator or an acknowledge signal indicating that the media content delivered from another computer system in network 1700 was successfully received. In one
5 embodiment, an acknowledge signal received by administrative node 1770 can generate a transaction applicable to the computer system receiving the media content and which is associated with the media content that was received.

Still referring to Figure 18, UCM 1800 further includes a secure player
10 application 1810 that can be configured to access contents of a protected media container file, (e.g., an instance of media) and provide presentation of the media content to the computer system on which the media content is stored. Secure player application 1810 can be further configured to utilize its access to the protected media container file and make available to the network the contents thereof. In one embodiment, secure
15 player application 1810 can present the media content to the computer system on which it is operable while the computer system is offline, (e.g., not coupled with network 1700 but still associated therewith).

Secure player application 1810 can be media type specific, such as a secure audio
20 player application for audio files, a secure video player application for video files, a secure alphanumeric application for text files, a secure software player application for software files, and so on. It is noted that secure player application 1810 can be configured to interact with nearly any media type.

In one embodiment, secure player application 1810 can be a custom media device 310 which can be an emulation of a custom media device driver (e.g., 307) as described herein with reference to Figures 3, 4, and Figures 5A-5D. In another embodiment, secure player application 1810 can be an approved player application as described
5 herein with reference to Figure 3.

Figure 19 is an exemplary system 1900 for controlling media sharing among multiple computer systems communicatively coupled in a network in accordance with one embodiment of the present invention. System 1900 includes client node 1705, a
10 source node 1715, and an administrative node 1770. Nodes 1705, 1715 and 1770 are implemented in and communicatively coupled to a network 1700 in one embodiment of the present invention. In one embodiment, network 1700 is a decentralized distributed network, (e.g., network 800 of Figure 8).

15 In an embodiment, source node 1715 and client node 1705 have logged on to network 1700, as described herein with reference to Figures 4, and steps 702 - 716 of Figure 7A. Client node 1705 requests an instance of media from network 1700, (e.g., 4321), shown as communication 1910. Network 1700 transfers the request to administrative node 1770 which determines the location of an instance of media 4321.
20 Administrative node 1770 responds to client node 1705 with the location of instance of media 4321, shown as communication 1920.

In this example, administrative node 1770 is cognizant that source node 1715 has an instance of media 4321 disposed therein. In the present embodiment, instance of
25 media content 4321 is stored in a protected media container file by memory coupled to

source node 1715. Additionally, instance of media 4321 is encrypted local to source node 1715, shown as dotted line 1716 encompassing media 4321.

Within Figure 19, source node 1715 sends a request to administrative node 1770
5 for an intermediate encryption key to encrypt instance of media 4321 for transfer to client node 1705, shown as communication 1930. Administrative node 1770 transmits an intermediate encryption key, (e.g. key 1775), to source node 1715. Upon receipt of intermediate encryption key 1775 by source node 1715, a client communication application 1820 operable on source node 1715 decrypts media 4321 from its encryption
10 local thereto, (e.g., encryption 1716), and encrypts media 4321 into an intermediate encryption, as indicated by dotted line 1775 encompassing media 4321.

Still referring to Figure 19, once client communication application 1820 on source node 1715 encrypts media 4321 into intermediate encryption 1775, client
15 communication application 1820 transmits media 4321 in intermediate encryption 1775 to client node 1705, shown as communication 1950. Upon client node 1705 receiving media 4321 in intermediate encryption 1775, client node 1705 sends a request to administrative node 1770 for an intermediate decryption key (e.g., key 1776), shown as communication 1960. Intermediate decryption key 1776 enables a client communication
20 application 1820 operable on client node 1705 to decrypt media 4321 from its intermediate encryption 1775. It is noted that until intermediate encryption 1775 is decrypted, secure player 1810 operable on client node 1705 is unable to present media 4321 to system 1705. It is further noted that if another computer system in network 1700 has intercepted media 4321 during transfer, the intermediate encryption 1775 prevents
25 use of media 4321 by the intercepting computer system.

Continuing, administrative node 1770 responds to client node 1705 request for an intermediate decryption key and transmits key 1776 to client node 1705, shown as communication 1970. Once key 1776 is received by client node 1705, a client communication application 1820 operable on system 1705 decrypts media 4321 out of intermediate encryption 1775 and encrypts media 4321 into an encryption local to client computer system, (e.g., encryption 1706 encompassing media 4321).

When media 4321 has been successfully received by and encrypted into an encryption local to client node 1705, (e.g., media 4321 with encryption 1706), client node 1705 transmits an acknowledge indicator indicating successful receipt of media 4321, shown as communication 1980. Accordingly, upon receipt of an acknowledge indicator, administrative node 1770 generates a transaction applicable to client node 1705 and which is associated with media 4321.

Advantageously, embodiments of the present invention provide components that enable controlled media sharing in a decentralized distributed network (a p2p network), e.g., network 800 of Figure 8 and/or network 1700 of Figure 17. Further advantageous is that embodiments of the present invention can also track the sharing, generate royalties applicable to a receiving computer system and associated with a particular instance of media. Also advantageous is that embodiments of the present invention allow for outside instances of media to enter the network and provide compliance with copyright restrictions and licensing agreements associated with the instance of media.

It is noted that one or more embodiments in accordance with the present invention described herein can be utilized in combination with a delivery system similar to delivery system 800 of Figure 8. For example, any network of one or more of the embodiments in accordance with the present invention can be substituted with a delivery system similar to delivery system 800. Alternatively, a delivery system similar to delivery 800 may be implemented to include any combination of source nodes (e.g., 1715), client nodes (e.g., 1705), and/or administrative nodes (e.g., 1770) that operate in any manner similar to that described herein, but are not limited to such.

MACINTOSH OPERATING SYSTEM

Figure 20 is an exemplary logic/bit path block diagram 2000 showing utilization of a copyright compliance mechanism 300 (of Figure 3), for selectively controlling recording of copyrighted media received by a Macintosh® (sometimes referred to as Apple®) computer system, (e.g., system 210), in one embodiment of the present invention. Copyright compliance mechanism (CCM) 300 is, in one embodiment, installed and operational on Macintosh® operating system in the manner described herein. Additionally, all rules and issues described herein regarding the CCM 300 and the media being controlled by the CCM 300 mechanism are enforced. The only difference is the method in which the CCM 300 apparatus is organized on the Macintosh® operating system (O/S). Although in one embodiment, CCM 300 is installed on the Macintosh® operating system, in another embodiment, the CCM 300 may be invoked by the Macintosh® operating system through a plurality of methods including system and periphery multimedia rendering applications (e.g., an iTunes player®, a CD player, a DVD player, a downloaded multimedia application, an MP3 player, or the like.)

As an overview, in one embodiment, the Macintosh® operating system receives the incoming media 499 from a source such as server 250. The incoming media 499 passes through the client main thread 2002 and is received by the multimedia framework 2025. From that point, a codec 2020 that knows how to decrypt the MPB encrypted incoming media 499 is activated. The codec 2020 will decrypt the incoming media 499 and then pass the decrypted data to a codec 2021 for transcoding and then passed to rendering software (e.g., AAC, MP3, WMA, or the like). The multimedia framework 2025 will then pass the data to the carbon framework 2022 or directly to the audio framework 2045 (audio HAL) for playback to the currently selected output device. The Audio HAL 2045 will transfer the audio data across the user 2080/kernel 2090 boundary, adding it into the IOAudioStream mix buffers. In general, IOAudioStreams represent the channels of an IOAudioEngine. IOAudioEngines represent a configuration of an IOAudioDevice and IOAudioDevices represent a physical output device. Each IOAudioStream will format and clip its mix buffer into its sample buffer. The IOAudioEngine object negotiates with a specific protocol, e.g., USP, FireWire, or a private protocol, to transfer the sample buffers to the device. The samples are transferred over the appropriate physical layers and played.

In general, the copyright compliance mechanism 300 of Figure 3 is broken into three parts on the Macintosh® operating system of Figure 20. In one embodiment, the three parts include a framework 2010, a Codec 2020, and a kernel extension 2030. Although three portions of the copyright compliance mechanism are described, there may be more or fewer portions on the Macintosh® operating system. For example, in one embodiment, there may be a plurality of frameworks 2010. In another embodiment, there may be no kernel extension 2030. In yet another embodiment, there

may be a plurality of kernel extensions 2030, each monitoring a single (or a plurality of) driver(s) (e.g., drivers 2041-2044). In yet another embodiment, there may be more than one Codec 2020, or there may be more than one media layer abstraction (e.g., multimedia framework 2025) monitored by a single Codec 2020.

5

Referring still to Figure 20, framework 2010 is similar in function to that of the application API's of CCM 300 described in detail herein. In general, frameworks are used in a Macintosh® environment to control various applications which may not be accessible at the kernel level. Frameworks for the Macintosh® environment are well known in the art. In one embodiment, the framework 2010 is attached to a Macintosh® operating system on a client computing system (e.g., system 210) to implement the MPB interface 2005. In one embodiment, the interface includes register, unregister, enable, disable, and update calls.

15 Register of framework 2010 occurs when the application informs the operating system (O/S) of its functionality. At register, the framework 2010 will ensure that all the other components of the CCM 300 are valid (e.g., that none have been spoofed, added, changed, removed, or the like). In addition, the framework 2010 will check to ensure that all the components of the CCM 300 are up to date and that a bad boy list is updated if necessary. The framework 2010 will then ensure that no bad boy processes are running (as described herein). In one embodiment, the framework 2010 will then verify that all relevant frameworks 2010 are in the expected locations and are valid. The framework 2010 then opens an initial instance of multimedia framework component 2025 which will stay open until unregister. If during the register process a compromise of the CCM 300 system is found, all components will be informed. In addition, in one

embodiment, if a compromise of the CCM 300 system is found, the framework 2010 will decide whether to shut down any media playback operation with or without warning.

With reference still to Figure 20, in one embodiment, at enable a monitoring
5 thread will be created. Monitoring threads are independent pieces of code well known in the art. As described herein, the monitoring thread will keep a constant eye out for illegal input devices (e.g., input devices banned by CCM 300 protocols as detailed herein). In addition, the monitoring thread will periodically interact with kernel
10 extension 2030 to obtain lower level information. If the monitoring thread finds any illegal input devices, the thread will pass back a message to framework 2010. As described in detail herein, framework 2010 will then disable the media playback operation with or without warning. In general, the monitoring thread is similar in function to monitor.exe described herein. The monitoring thread may be updated at start-up, during a restart, or may be updated after start-up with changes to take effect
15 after the next restart.

The framework 2010 may disable the media playback utilizing any of the methods described in detail herein. For example, in one embodiment, the framework 2010 may inform Codec 2020 to disable the media playback capabilities (to include
20 encryption/decryption capabilities) of the media player (e.g., Quicktime®, iTunes player®, or other Macintosh® O/S media players) thereby stopping output of the controlled media.

In general, the framework 2010 in conjunction with Codec 2020 will be able to
25 detect any application with taps into the audio framework 2045 (e.g., illegal applications

2050 such as audio hijack, or the like) at the user level 2080 and block their capabilities. For example, if the framework 2010 detects an illegal application 2050 trying to access the core audio framework 2045, the framework 2010 informs the codec 2020 to disable the media playback (e.g., stop decryption, or the like). In another embodiment, the
5 framework 2010 will initiate a gate 2055 between the illegal application 2050 and the core audio framework 2045 thereby preventing the illegal application 2050 from accessing the media. In general, the gate 2055 may be any type of switch, disconnect, etc. such as those described in detail herein.

10 Referring still to Figure 20, the kernel extension 2030 is used to thwart kernel level recorders 2070 (e.g., wiretap) by using an IOProvider IOResources which loads at startup and remains open permanently even though it matches no physical device. In general, kernel extension 2030 presents a standard IOUserClient interface that will allow any user space process to communicate with it through a private API. In one
15 embodiment, the kernel extension 2030 will provide unfettered access to the machine's resources and the kernel data space. In one embodiment, the kernel extension 2030 includes a kernel level driver (or a plurality of kernel level drivers) which may be turned on or off in contingent with the business rules described in detail herein. By controlling media framework 2025, copyright compliance mechanism 300 can prevent
20 unauthorized output of incoming media 499 (of Figure 4).

With the addition of the kernel extension 2030, the kernel level sound pathways will be protected by kernel extensions which may be small drivers (e.g., the Macintosh® version of the shims described in detail herein) operating at the kernel level 2090. For
25 example, the drive speaker 2041 may be unhindered, but the kernel extension 2030 may

use drivers at the driver microphone 2042 (e.g., the analog hole described herein), the driver full duplex 2043, the driver Ipod® 2044, and any other output drivers within the system to control or block output. For example, the kernel extension 2030 small drivers may provide gates (e.g., 2062, 2063, and 2064) which will stop access to the media if the
5 business rules described herein necessitate it. For example, depending upon the copyright restrictions and licensing agreements applicable to an incoming media file, (e.g., 499), kernel extension 2030 controls whether gates 2062, 2063, and 2064 are open (shown) thereby preventing incoming media 499 (of Figure 4) from reaching a media recording application, or closed (not shown) to allow recording of incoming media 499.
10 In addition to utilizing drivers to control gates to access known output drivers within the system, in one embodiment, kernel extension 2030 also includes a kernel driver which may be applied to a driver previously unnamed. For example, once the kernel level recorder 2070 (e.g., wiretap or other illicit Macintosh® O/S driver) is recognized (e.g., by monitoring thread, kernel extension 2030, or the like) the kernel extension 2030
15 will prevent access from the kernel recorder 2070 via a gate 2075.

In one embodiment, as described herein, copyright compliance mechanism 300 is shown as being communicatively coupled with a content server, e.g., 251. As described in detail herein, in one embodiment, incoming media 499 may originate from content
20 server 251. In another example, incoming media 499 may originate from a personal recording/electronic device, (e.g., a MP3 player/recorder or similar device), coupled to system 2000. Alternatively, incoming media 499 may originate from a magnetic, optical or alternative media storage device inserted into a media device player coupled to system 2000, (e.g., a CD or DVD inserted into a CD or DVD player), a hard disk in a hot
25 swappable hard drive, an SD (secure digital card) inserted into a SD reader, and the

like. In yet another example, incoming media 499 may originate from another media player application or media recording application. Incoming media 499 may also originate from a satellite radio feed (e.g., XM radio), a personal communication device (e.g., a mobile phone), a cable television radio input, e.g., DMX (digital music express), a digital distribution and/or a public presentation source via a network, Internet or other communication connection, pay-per-view and/or pay-per-play system, or a set-top box. It is noted that incoming media 499 can originate from nearly any source that can be coupled to system 2000. Regardless of the source of incoming media 499, embodiments of the present invention will prevent unauthorized recording of the media 499.

With reference now to Figure 21, a flowchart 2100 of a process for preventing unauthorized recording of media on a Macintosh® operating system is shown in accordance with one embodiment of the present invention. In one embodiment, the media is from a source coupled with the client system, wherein the source is from the group consisting of: a network, a personal communication device, a satellite radio feed, a cable television radio input, a set-top box, an media device, a media storage device, a media storage device inserted in a media device player, a media player application, and a media recorder application.

Referring now to step 2102 of figure 21 and to Figure 20, in one embodiment a compliance mechanism (e.g., CCM 300) is registered on a client system having the Macintosh® operating system operating thereon. As described herein, the CCM 300 may be installed via a network download (e.g., from server 250), a CD, a Floppy disk, a DVD, or any other media which may be inserted, downloaded, or received by the computing system. As described herein, the CCM 300 may include a plurality of

business rules (e.g., copyright, sharing, access, etc.). In addition, in one embodiment the CCM 300 business rules may be altered in response to a change in a usage restriction. For example a usage restriction comprising a copyright restriction or licensing agreement applicable to the media.

5

With reference now to step 2104 of Figure 21 and to Figure 20, in one embodiment, the CCM 300 includes a framework 2010 which is used for validating the CCM 300 on the operating system 2000. In one embodiment, the framework 2010 will disable audio playback from the multimedia component until the components of the compliance mechanism are validated. Additionally, the framework 2010 accesses a network (e.g., server 250) to ensure that the components of the CCM 300 are up to date. Furthermore, the framework 2010 establishes a monitoring thread which maintains a constant search for output devices.

15

Referring now to step 2106 of Figure 21 and to Figure 20, in one embodiment, a multimedia component for decrypting the media content on the Macintosh® operating system is opened by the framework 2010. As described herein, the multimedia component 2025 may be a media rendering or processing application such as a Quicktime® framework, iTunes player®, or the like.

20

In one embodiment, the CCM 300 may further include a kernel level extension 2030 providing kernel level driver information to the framework 2010, wherein when an invalid kernel level extension is recognized the framework 2010 directs the kernel level extension 2030 to selectively restrict output of the media content. In one embodiment, the kernel level extension 2030 matches no physical device on the client system. In

general, the kernel level extension 2030 recognizes a kernel level recorder capturing an audio stream and informs the framework 2010 of the kernel level recorder 2070 (e.g., wiretap or the like). The CCM 300 may further comprises a bad boy list such as those described herein.

5

With reference now to step 2108 of Figure 21 and Figure 20, the decryption of the media content on the client system having the Macintosh® operating system operating thereon is prevented if a portion of the CCM 300 is invalidated.

10

Thus, embodiments of the present invention provide a method that prevents unauthorized recording of media files on a Macintosh operating system. Further, embodiments of the present invention provide a method that prevents unauthorized recording of media files on a Macintosh operating system that can be updated over a network. Embodiments of the present invention also provide a method that prevents unauthorized recording of media files on a Macintosh operating system which is compatible with industry standards.

15

The foregoing disclosure regarding specific embodiments of the present invention has been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and many modifications and variations are possible in light of above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited

20

to the particular use contemplated. It is intended that the scope of the invention be defined by the Claims appended hereto and their equivalents.